DYNAMICS AND MOTION CONTROL OF UNDERWATER REMOTELY OPERATED VEHICLES AND HIGHLY FLEXIBLE ELASTIC RODS

JUAN ALBERTO RAMÍREZ MACÍAS

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
DOCTORADO EN INGENIERÍA
MEDELLÍN
2019

DYNAMICS AND MOTION CONTROL OF UNDERWATER REMOTELY OPERATED VEHICLES AND HIGHLY FLEXIBLE ELASTIC RODS

JUAN ALBERTO RAMÍREZ MACÍAS

Dissertation submitted to
the School of Engineering in partial fulfilment of the requirements
for the degree of Ph.D. in Engineering (Doctor en Ingeniería)

Advisor(s)
Rafael E. Vásquez, Ph.D.
Asgeir J. Sørensen, Ph.D.
Svein Sævik, Ph.D.

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
DOCTORADO EN INGENIERÍA
MEDELLÍN
2019

DECLARACIÓN DE ORIGINALIDAD

August 22, 2019

Juan Alberto Ramírez Macías

“Declaro que este trabajo de grado no ha sido presentado con anterioridad para optar a un título, ya sea en igual forma o con variaciones, en ésta o en cualquiera otra universidad”. Art. 92, parágrafo, Régimen Estudiantil de Formación Avanzada.

Firma:

Juan Alberto Ramírez M.

To the memory of Hugo, I wish one day my heart gets as big as yours.

To Nubia, I wish one day my soul gets as strong as yours.

To Oso, every day you make me prouder.

To Jenny, thank you for bringing me back to Earth.

Abstract

This thesis develops analytical- and numerical-oriented tools for the dynamic analysis and motion control of ROVs, using an integrated approach where hydrodynamics, control, and cable effects are considered. These tools facilitate the analysis, design, and operation of ROV systems and are aimed at answering questions about their motion feasibility. They are grounded on available theory to model the ROV and cable, design of feedback control, and thrust allocation. Subsequently, this work contributes means to study the hydrodynamics of an ROV and develops a framework where hydrodynamics and other motion-relevant systems can be integrated and evaluated. For studying hydrodynamics, the manoeuvring behaviour of ROVs is approached by using time-domain simulation (TDS) and computational fluid dynamics (CFD). In this approach, viscous-flow computations are used to gather data about the hydrodynamic forces and moments that act on an ROV at certain flow conditions, and from that data a simplified meta-model is assembled to be used in time-domain simulations for manoeuvring, control, and motion feasibility analyses. For integrated motion feasibility studies, the concept of dynamic positioning capability (DPCap) for ROVs is proposed. The idea of DPCap was first introduced for surface vessels that include a dynamic positioning (DP) system, and in this work it is extended to ROVs. DPCap assesses whether craft are able to withstand environmental and operational loads, while keeping a desired position or path. This allows one to determine whether the system will be able to operate or not at certain conditions and quantify the level of motion capability. Through these approaches, this work delivers a framework to answer questions about ROV motion feasibility joining vehicle dynamics, hydrodynamics, cable mechanics, and control, and that intends to be useful at different stages of an ROV's life-cycle.

KEYWORDS:

Cable modelling; Computational Fluid Dynamics (CFD), Design of Computer Experiments (DoCE); Dynamic Positioning Capability (DPCap); Hydrodynamics; Manoeuvring; Remotely Operated Vehicles (ROV); Time-domain Simulation (TDS)

ABBREVIATIONS

AMOS: [Centre for] Autonomous Maritime Operations and Systems
AUV: Autonomous Underwater Vehicle
CFD: Computer Fluid Dynamics
Colciencias: Departamento Administrativo de Ciencia Tecnología e Innovación (Colombian Department of Science, Technology, and Innovation)
CTD: Conductivity-Temperature-Depth [sensor]
DoCE: Design of Computer Experiments
DoE: Design of Experiments
DP: Dynamic Positioning
DPCap: Dynamic Positioning Capability
DVL: Doppler Velocity Log
EFD: Experimental Fluid Dynamics
FD: Finite Differences
FEM: Finite Elements Method
LH: Latin Hypercube
LHD: Latin Hypercube Design
LHS: Latin Hypercube Sampling
MARIN: Maritime Research Institute Netherlands
MCD: Monte-Carlo Design
MCS: Monte-Carlo Sampling
MSROV: Mid-Sized ROV
NED: North-East-Down [reference frame]
NTNU: Norwegian University of Science and Technology
OCROV: Observation Class ROV
ODE: Ordinary Differential Equation
OHD: Orthogonal Hypercube Design
PDE: Partial Differential Equation

PID: Proportional-Integral-Derivative [control]
PMM: Planar Motion Mechanism
RANS: Reynolds-Averaged Navier-Stokes [equations]
ROV: Remotely Operated Vehicle
TDS: Time-domain Simulation
TMS: Tether Management System
UNALMED: Universidad Nacional de Colombia, sede Medellín.
UPB: Universidad Pontificia Bolivariana
VISOR: Vehículo para Investigación Subacuática Operado Remotamente (Remotely Operated Vehicle for Underwater Research)
VTC: Virtual Captive Test
WCROV: Work Class ROV

Acknowledgements

FORMAL, FUNDING-RELATED ACKNOWLEDGEMENTS

The author's doctoral studies were supported by funds of the Universidad Pontificia Bolivariana (UPB), the Strategic Program for the Development of Robotic Technology for Offshore Exploration of the Colombian Seabed, project 1210–531–30550, contract 0265–2013, and the author. The Strategic Program for the Development of Robotic Technology for Offshore Exploration of the Colombian Seabed has been developed with the funding of the Fondo Nacional de Financiamiento para la Ciencia, la Tecnología y la Innovación, Francisco José de Caldas; the Colombian petroleum company (ECOPETROL); Universidad Pontificia Bolivariana; and Universidad Nacional de Colombia—Sede Medellín (UNALMED). This work was also partly supported by MARIN Academy and MARIN's Offshore Department, which sponsored the research project Manoeuvring study of an ROV using CFD and time-domain simulations, and the Research Council of Norway through the Centres of Excellence funding scheme, project number 223254—AMOS. The author thanks MARIN Academy for the author's internship, carried out in Fall 2015, and the Norwegian University of Science and Technology's Centre for Autonomous Marine Operations and Systems for the author's internship during Fall 2016 and the short stay during Summer 2017.

THE PEOPLE AND CIRCUMSTANCES THAT MADE THIS TO BE BORN

I officially became a Ph.D. student during mid-2015. I had discarded the possibility of ever becoming a Ph.D. because my professional life was taking me to a rather different path; moreover, being over 30, I felt rather old to start something like that (and I was partly right). Perhaps this story started when, while being part of the *Strategic Program for the Development of Robotic Technology for Offshore Exploration of the Colombian Seabed*, we visited different research institutions in Europe (including MARIN and NTNU's AMOS): the possibility of collaborating with MARIN and AMOS made it reasonable to have a Ph.D. student. Later, I was chosen to walk that road: I have to thank Rafael for that. This idea became a reality thanks to different people in UPB, from which I acknowledge Rafael that later became my advisor; Julio, main researcher of the program; Diego, head of Mechanical Engineering; and Piedad, dean of the School of Engineering at that time. Thanks to them, I got the necessary funding that made all possible. I thank Julio for vouching for me financially, a not-so-easy-to-find requisite for the scholarship with UPB.

THE PEOPLE WHO ADVISED THE WORK

I thank Rafael for advising this thesis and trusting in my work. Thanks for reminding me constantly that a Ph.D. must be finished and, consequently, steering the work in that direction.

I thank Asgeir for co-supervising this thesis: I learned from you a lot. I treasure fondly the conversations we had. Being host at AMOS was a privilege; those days in Norway were brutally learning-intensive. I thank Svein for co-supervising the cable part of this thesis. It was nice to discuss the *bloody details*; those discussions were quite enlightening.

I thank Persijn for supervising my internship at MARIN and opening the door to me to the world of craft manoeuvring. I thank Tomasz and Chris for teaching me how to do CFD in ReFRESCO. I thank Serge, whose work was this thesis' hydrodynamics spinal cord. And, finally, I thank Timo for opening the doors to us at MARIN; thank you as well for the copy of Kuiper's Wageningen Propeller Series book (a treasure I still keep).

THE PEOPLE AT THE UNDERWATER ROBOTICS PROGRAM

This work was part of the *Strategic Program for the Development of Robotic Technology for Offshore Exploration of the Colombian Seabed.* This project was ambitious and, to our abilities, quite difficult: those growth-by-force types. Life around the project, and mixed with the studies, was an intricate life experience: there was too much to learn professionally and personally. From the project, I thank the team of researchers Elkin, Diego, Carlos, Norha, and Jorge. I thank as well the graduate students Checho, Luis Mi, Andrea, Esteban (Palacio), Cindy, Diana, Yesid, and Cristian: we worked for a short time, but you gave me perhaps the most rewarding professional experience I have had so far. A huge thank you to Esteban (Vargas), José, and Juan Camilo: advising your Master's was a honour, you allowed me to understand many things better.

THE PEOPLE I MET FAR FROM HOME

The time I spent abroad was relatively short, but for me it was intense. I met many people who were able to impact my life positively, regardless of the small time we had together. From the Netherlands time I remember fondly Antoine, Hans, Rinske, Alexandre, Yang, Nicola, Just, Sarah, Swaraj, Carlo, Nilakash, Giorgio, Diana, Paolo, and Victor; and from the time in Norway, Ricardo, Mauro, Alexandra, Ida, Albert, Sepideh, Kasper, Thomas, Martin and Giova, Wilson, Martin, Pedro, Christoph, Astrid, Jeevith, Erlend, Mikkel, Michel, Andreas, Jørgen, and Annika. I hope I didn't leave anyone outside.

FRIENDS AND COLLEAGUES

Thank you to my friends and colleagues. To Elkin and Diego: thank you for lending me your ears; you kept me standing. To Cristian: thank you for being close during the years. To David and Juanita, Edwin and Sandra, and Alejo: I keep you in my heart; I'm fortunate to be able to count on you; I just admire you. To Camilo and Laura: thank you for giving me a home in Eindhoven. To the colleagues/friends along the way, Juan Miguel, Rozo, Raúl, Marisol, Manolo, Gabriel, and Juan Daniel: thank you.

♡

The hugest thank you goes to my family. Mom, thank you for being that force of nature. Grandma, thank you for bringing family together. Oso, thank you for keeping your heart always warm. Jenny, thank you, just thank you, you were there all the time, despite the incredible beasts; you're my not-imaginary wonderwall. Dad, how I wish you were still with us, thank you, you are my beacon, your memory is indelible.

Contents

Pag.

List of Figures

Pag.

List of Tables

Pag.

Chapter 1

Introduction

1.1 MOTIVATION

This work is motivated by the long-term initiative to create technological tools that enable the development of an ocean-based sustainable economy. The idea of a sustainable ocean economy is often called *The Blue Economy* by many private and governmental organisations. A blue economy balances the fulfilment of humanity's needs and the ocean's ability to stay healthy on a large time scale [91, 42]. The ocean is often seen as a way to cope with humanity's resource-related issues, but, given the fact that the ocean's health has been already altered [58], the overall increase of human activity raises concerns and requires taking action towards sustainable use of marine resources [137]. As a country with roughly half its territory in the ocean, this concept is pertinent to Colombia.

As a plausible approach, the general goal of developing a blue economy may be tackled by increasing knowledge, *i.e.*, knowing the ocean's environment better (*e.g.* as proposed in [59]) and developing technology intelligently. Increasing knowledge about the environment allows the assessment of both the amount of resources and impact of human interventions, and better-developed technologies are targeted to be more effective, efficient, and safe. By and large, technology development plays a crucial role, because it provides the means to safely explore and study the ocean's environment and allow human activities.

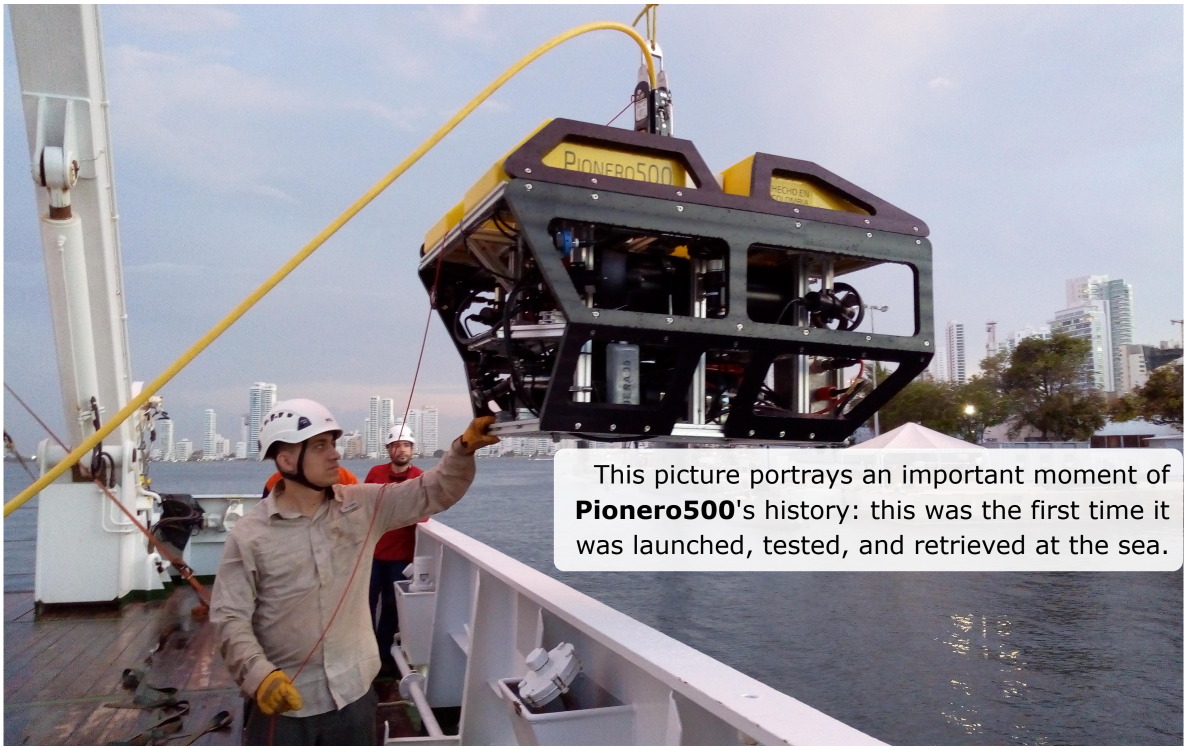
Tasks such as mapping and monitoring the environment may be accomplished by using different autonomous or intelligent equipment [78]. According to Nilssen *et al.* [86], each platform becomes useful depending on time and space scales. Such platforms may be fixed or mobile, and among the mobile one can find remotely operated vehicles (ROV), autonomous underwater vehicles (AUV), and gliders. Even though in the scientific and technical community myriad efforts are being directed towards more autonomous platforms, many underwater operations still require the use of ROVs intensively; the amount of power and data bandwidth pose limitations to untethered systems, and many operations still need real-time human supervision.

More technically, ROVs are self-propelled submersible robots that are connected through a cable (tether or umbilical) to a remote control station for real-time operation and/or supervision [26]; see *e.g.* Fig. 1.1. There are two features that are common to ROVs: first, they are primarily designed to be rather slow mobile platforms for the deployment of tools and sensors into specific surveying sites; and, second, they require any sort of cable link that connects them to a surface vessel or any other station for providing control and/or supervision in real time and, often, power as well. Typically, operations with ROVs are performed by either having a cable that connects the vessel and the vehicle directly or through a tether management system (TMS). In the latter case, a TMS works as an intermediate stage, where a garage containing the ROV is deployed from the vessel to the operation zone and, later on, the ROV is deployed from the garage to explore the nearby area.

An approach to perform better or more intelligent operations is to consider the problem of properly manoeuvring all or most of the mobile components. For an ROV those are essentially the vehicle and cable, although it is possible to consider a surface vessel or any other mobile components as well. Manoeuvring an ROV-cable system may be seen as a two-fold problem: considering the hydrodynamic effects of the medium and taking into account the motion of the cable. In the former, the hydrodynamic effects are to be considered both in the cable and vehicle but are more complex in the vehicle. In the latter, the motion of a cable is that of a rod type of slender structure, *i.e.*, a long, highly deformable body, whose length is many orders of magnitude larger than any other dimension (*e.g.* the diameter). An ROV cable has similar attributes to those of other ocean engineering systems components, such as power and umbilical cables and tethers, as well as pipes, risers, and the like.

Considering, for instance, an operation where the ROV is deployed directly from the vessel, the motion of the cable is influenced mainly by the motion of the vessel and ROV and the surrounding environmental conditions, *e.g.*, current and waves. In this case, as well as in other cases, the cable is not often directly modelled and treated as a disturbance. Not having a more elaborate scheme to consider the cable mechanics more accurately limits the number of possible analyses, and, furthermore, the types of underwater activities that may be accomplished.

An alternative to improve existing and foresee new conceptual design solutions and operation



This picture portrays an important moment of **Pionero500**'s history: this was the first time it was launched, tested, and retrieved at the sea.

Figure 1.1. Pionero500, ROV developed at Universidad Pontificia Bolivariana, during deployment from the vessel to the ocean. This picture was taken by the author in Cartagena Bay, Colombia, 2018. In the picture are Elkin Taborda (front) and Luis Miguel Aristizábal (back), part of the developer team.

schemes, where ROVs and elastic rods' manoeuvring is considered, is by applying modelling and numerical simulation techniques to analyse and design operation and control scenarios. Although mathematical models do not necessarily predict the complete spectrum of phenomena, a good-enough model allows one to opportunely plan and predict a numerous amount of situations. In a broader sense, a model that makes good predictions reflects knowledge maturity about the system that is being studied and reduces the levels of uncertainty. Therefore, in this work, an integrated approach for modelling, simulation, and control of ROVs and rod-like structures is proposed as an alternative for analysing and designing ROV systems and operational schemes for accessing the ocean space effectively, efficiently, and safely.

1.2 BACKGROUND

1.2.1 ROV TECHNOLOGY IN UPB

Since the 1990's, researchers at the Universidad Pontificia Bolivariana (UPB) in Medellín, Colombia, have been developing underwater craft for exploring underwater. Motivations, technologies, and people have varied during the decades, but the interest in underwater exploration has survived the test of time. The first ROV prototype, developed during the 1990's, was named VISOR after an acronym for remotely operated vehicle for underwater research (*vehículo para investigación subacuática operado remotamente*). This prototype was first developed to investigate sounds produced by whales during migration across the Colombian Pacific Ocean. The second prototype, developed during the early 2000's was named VISOR II. The motivations behind that second vehicle were more technologically related. The vehicle was shaped using a streamlined geometry and the control scheme was intended to have dual operation: remotely operated or autonomous. There are not much records of these first two developments but their history has been reported in further work [100].

The third prototype was developed during the late 2000's and was called VISOR3 (see the smaller vehicle in Fig. 1.2). This vehicle was initially designed and developed for surveillance and inspection of port facilities [100, 57]. Following the initial development, and more intensively since 2015, it has been used as a test-bed for studying ROV-related subjects such as hydrodynamics [131, 98], low- and high-level control and perception algorithms [147, 8], development of embedded control systems and user interfaces [7, 106, 56, 92], and design architectures [31]. A lot of what we know today about ROVs has been accomplished through modelling, simulation, and experimentation on VISOR3. Recently, VISOR3 was used as equipment during the Seaflower Scientific Expedition 2016 [133]. All VISOR prototypes may be regarded as observation class ROVs, given their small size and observation-only functionalities (see Sec. 1.2.3 or refs. [26, 20] for information about ROV classes).

The last prototype so far has been a mid-size ROV called PIONERO500 (see the bigger vehicle in Fig. 1.2 or Fig. 1.1), a prototype that has brought greater challenges related to its bigger scale and greater complexity [30]. This prototype has been developed as part of the *Strategic program for the development of robotic technology for offshore exploration of the Colombian Seabed*, funded by the Fondo Nacional de Financiamiento para la Ciencia, la Tecnología y la Innovación, Francisco José de Caldas (Colciencias), Colombian petroleum company (Ecopetrol), Universidad Pontificia Bolivariana—Medellín (UPB), and Universidad Nacional de Colombia—Sede Medellín (UNALMED). PIONERO500 has made us push our

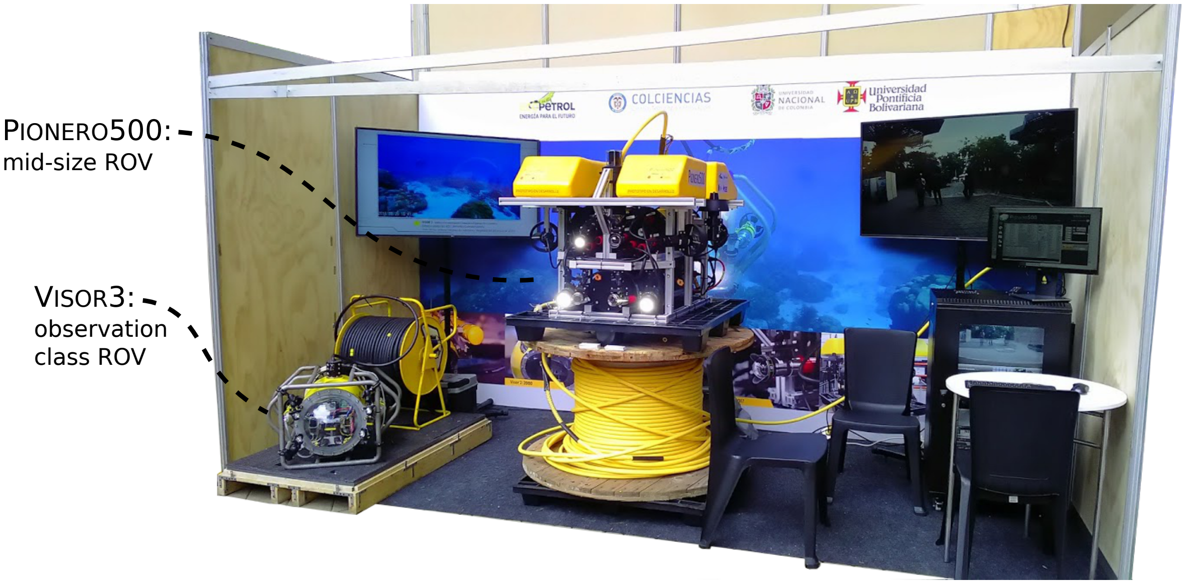


Figure 1.2. A small glance at UPB's recent ROV history: Pionero500 and visor3 in Senalmar 2017. It is not evident, but in the background pictures UPB's oldest ROVs were shown: VISOR and VISOR II. The picture was shot by Esteban Palacio and edited by the author.

knowledge boundaries even further.

1.2.2 THIS THESIS' BACKGROUND

Dating back to this work's own history, since the first stages, the idea in mind was to join knowledge from hydrodynamics and control, to be useful during ROV design. This idea followed the heuristics that hydrodynamics people focus on optimising geometries and control people do not see geometry as the main priority. Then, maybe, finding common ground for hydrodynamics and control may make possible to address such issues early during the design process: this has been conjectured and experienced as not trivial. Moreover, hydrodynamics and control share the feature that both require very precise information to perform useful computations. For instance, to analyse drag performance through computational fluid dynamics (CFD), one requires a virtual three-dimensional geometry and through experimental fluid dynamics (EFD) a real three-dimensional model. Conversely, to design a control system one requires at least a mathematical model of the dynamics. All this information is often only available at the later stages of the design or during the manufacturing process. During Pionero500's early design process (Universidad Pontificia Bolivariana's latest ROV development), it was cumbersome to answer technical, quantitative questions that involved hydrodynamics, cable mechanics, or control. This work got a great deal of motivation from those rather frustrating, unsuccessful experiences.

The initial idea of joining knowledge from hydrodynamics and control for ROV design resulted from various conversations with Rafael Vásquez in UPB. Afterwards, this work was heavily influenced by experiences gathered abroad in the Maritime Research Institute Netherlands (MARIN) and Centre for Autonomous Maritime Operations and Systems (AMOS) of the Norwegian University of Science and Technology (NTNU). The idea of using viscous flow computations for obtaining hydrodynamic loads (using the virtual captive test, VCT, approach) and then using the data to obtain a manoeuvring model was developed during an internship in MARIN. The base idea was originally presented in the thesis by Toxopeus [126] for ships and submarines, and the proposal of expanding it to ROV's was conducted by Persijn Brongers and advised by Tomasz Gornicz, Chris Willemsen, and Serge Toxopeus in MARIN. Subsequent work in UPB revealed that a manoeuvring would not be complete if cable mechanics is not considered: this motivated further work in AMOS-NTNU. As a result, much of this thesis' path was crafted during conversations with Asgeir Sørensen at AMOS-NTNU, such as how to approach the cable problem, as well as the idea of extending dynamic positioning capability analyses from surface vessels to ROVs. Further ideas related to cable modelling were discussed with Svein Sævik. Moreover, people at AMOS have gatherer a considerable amount of expertise tackling the problem of increasing ROV autonomy (*i.e.*, intelligence); therefore, this work is heavily influenced by the theses of Ludvigsen [77], Dukan [36], Fernandes [32], and Candeloro [20]. This work is grounded on the aforementioned theses and proposes its own particular path of study.

1.2.3 REMOTELY OPERATED VEHICLES

As mentioned in Sec. 1.1, remotely operated vehicles (ROV) are self-propelled submersible robots used for underwater inspection and intervention tasks. They are commonly used as platforms for the deployment of tools and sensors to specific survey sites. ROVs are considered mature technology, with a well-known history of being a handy tool for accomplishing from simple to highly dangerous tasks [26]. This technology has been widely used by the oil and gas industry but, more recently, has expanded to a larger span of applications such as renewable energies [47], sea-floor mining [2], aquaculture [108], biological studies [29], and archaeological exploration [90], among others. Future work and application prospects suggest that improving knowledge and capabilities around ROV technology, *e.g.*, increasing autonomy, is a necessary task [78].

ROVs have different levels of autonomy (see *e.g.* [20] for a discussion about autonomy) but are always commanded or supervised and often powered from the surface using a tether [26]. In terms of autonomy, ROVs are often compared to AUVs because the latter perform preprogrammed, autonomous operations and lack a tether. As an effort to combine features from both worlds, there are hybrid ROVs that aim to perform ROV-like tasks but without a tether. This work focuses on tethered ROVs. Operations with ROVs are commonly performed either having the vehicle directly deployed and tethered from a ship or using a tether management system (TMS). When using a TMS, a garage containing the vehicle is deployed into the water and then the vehicle is deployed from the garage. Spatial and temporal resolution and coverage of ROV operations range from a few centimetres to a few kilometres and monitor processes that change in the order of a few seconds to one day [86].

ROVs are usually designed to have enough motion capability to be a platform for the deployment of tools and sensors for surveying specific sites. Motion autonomy requires the use of navigation sensors such as acoustic baseline sensors, Doppler velocity log (DVL), pressure sensors, heading sensor, and inertial sensors such as compass and inertial measurement units (IMU) or motion reference units (MRU). Additionally, operations for ocean mapping and monitoring require deployment of sensors such as video cameras, underwater hyperspectral imaging (UHI), conductivity-temperature-depth (CTD) sensors, magnetometer, acoustic Doppler current profilers (ADCP), active sonars, and navigation-related sensors [78].

Depending on size, power, and operation capabilities, ROVs are often classified as observation class (OCROVs), mid-sized (MSROVs), and work class (WCROVs) [26]. OCROVs are the smallest vehicles; they often weigh less than 100 kg, have very limited payload and depth rating, *e.g.* UPB's VISOR3 (Fig. 1.2) is an OCROV. MSROVs may be considered an extended-capability version of the OCROVs. They are often bigger, heavier, and more powerful; have greater payload capacity and depth rating; and may carry light equipment for low-complexity tasks. For example, *e.g.* UPB's PIONERO500 (Fig. 1.2) is a MSROV. WCROVs are the heaviest and most powerful class; consequently, they run on hydraulic power and are capable of performing heavy work.

ROVs are an active research field. Two typical research approaches are the use of ROVs as components of multi-robot missions or the improvement of the ROV system itself. In the former case, joint operations of ROVs with AUVs are often studied [78]. In the latter case, because the type of tasks ROVs accomplish normally require human intervention in real time, a recurrent topic of research is how to improve ROV systems by increasing their operation autonomy and complexity. Examples include user-machine interaction [22] and navigation and real-time planning capabilities [21].

1.2.4 A NOTE ON DYNAMICS, THRUST ALLOCATION, AND CONTROL

The dynamics of ROVs is a well-studied subject derived from rigid body dynamics applied to water craft. A widely used formulation is Fossen's robot-like dynamic equation [48, 36]; the name is due to it resemblance to classical dynamics formulations used in robotics. These are compact, generalised equations compatible with Newton's second law that model craft behaviour in the forces and moments space as a function of position and velocity. There are other approaches that make use of ideas and notation coming from the field of geometric mechanics (see for instance [15, 88]). These are often based on variational principles directly, such as Hamilton's principle of least action, and the notion of a geometrical configuration space (or manifold) where motion lies. From a practical standpoint, the latter approach differs from the former by how numerical time-integration algorithms are derived.

Thrust allocation is the most basic action towards controlling an ROV, even when the vehicle is manually controlled and no feedback control strategy has been implemented. The task of thrust allocation is to translate motion commands in three-dimensional space into thruster forces [36]. Depending on how thrusters are distributed, the system may be under-actuated, fully actuated, or over-actuated. Thrust allocation is based on the idea that a group of actuators produce a net force and moment that is responsible of motion, and, conversely, one should be able to compute the inverse action: given a desired net force and moment, compute each actuator's required force.

Feedback control is often approached by using a non-linear PID scheme [32]. This approach looks forward to applying PID control's classical idea of computing proportional, integral, and derivative actions from an error to the particular context of water craft such as ROVs. These craft have the particularity that often one wants to control motion variables on a global x–y–z frame (such as the often-used North-East-Down, NED, frame) by manipulating forces and moments on a (different) frame that is attached to the body. The need of using a kinematic transformation between these frames makes the control law non-linear. This and other control laws depend on knowing all motion-related variables. This is accomplished by using an algorithm that uses (noisy) measurements and estimates a filtered version of known and previously unknown variables; often this estimator is based on the Kalman filter [48]. This work will further assume that such an estimator is constructable and all required variables are available; consequently, it will not deal with how to construct such estimator nor its effects on motion feasibility.

1.2.5 HYDRODYNAMICS AND MANOEUVRING

Manoeuvring studies are commonplace for ships, submarines, and autonomous underwater vehicles (AUV) and have been an active research subject for decades (see for example refs. [129, 48, 126, 23]). The motion of water craft is highly dependent on hydrodynamics. Historically, the study of hydrodynamics has a time-standing duality between theory and experimentation: today's knowledge is due to model testing and empirical formulas derived from it, as well as to pursuing practical computations of mathematical models derived from first principles [83]. Because the latter was only approachable in simple cases, experimental fluid dynamics (EFD) has been used as the main tool to acquire information about hydrodynamic behaviour of water craft, such as towing tank and planar motion mechanism (PMM) experiments [65]. After the development of computational fluid dynamics (CFD) software, numerical solvers have been used as the *de facto* tool for calculating theoretical models.

Although experimental testing is fundamental for 'true' calculation of hydrodynamic loads, there are a number reasons to choose theoretical/computational approaches. To name a few, testing facilities are not always available and often expensive to use, fabrication of physical models and assembling the experiment may be time-demanding, and so on. Consequently, CFD has been explored as an alternative to experimental fluid dynamics to speed-up the process and reduce experimental facilities' costs [127, 126, 70].

Well-known CFD computations and manoeuvring studies have been carried out. Vaz *et al.* [135] compared two viscous-flow solvers for the accurate prediction of the manoeuvring forces of the DARPA SUBOFF submarine. Zhang *et al.* [146] computed hydrodynamic coefficients for a long-endurance underwater vehicle. Toxopeus *et al.* [128] worked on the validation of tools to simulate the manoeuvrability and seakeeping of sea vehicles. Wang *et al.* [139] developed the mathematical model for an underwater vehicle based on CFD calculations, strip theory, and open-water tests. Furthermore, different combinations of experimental, computational and model regression techniques have been tested as well [89, 103, 5, 80, 123, 14].

Because ROVs are primarily designed to be rather slow mobile platforms for the deployment of tools and sensors, hydrodynamics is not commonly a concern as it is to other underwater vehicles. Nonetheless, this does not mean that hydrodynamics should be completely avoided. There have been works where different strategies are used to obtain ROV-shaped craft hydrodynamics models using experimental results and simulation. For instance, Tang *et al.* [124] studied the TUNA-SAND hydrodynamic model through different experiments and using FLUENT. Avila and Adamowski [9] and Avila *et al.* [10] determined hydrodynamic coefficients of full scale ROVs using a PMM installation, least squares regression, and system identification;

they also proposed a three-step identification procedure for such vehicles. Chin and Lum [25] proposed an xPC target platform in order to experimentally determine hydrodynamic parameters. Aras *et al.* [6] modelled a low cost ROV using a system identification toolbox. Chin and Lau [24] used CFD software for the systematic modelling of hydrodynamic damping of a complex-shaped ROV, by using ANSYS-CFX to model an ROV's damping model and a free-decay pendulum test for verification. Velasco *et al.* [136] performed a complete modelling of a torpedo-shaped underwater vehicle using parameter estimation with data acquired in the model basin. Xu *et al.* [145] obtained a hydrodynamic model from a test model considering the effects of the ROV's asymmetric shape. Despite these available works, from the author's perspective, there is still research to be done around the study of manoeuvring and hydrodynamic behaviour of ROVs.

As part of this thesis' preliminary works, methodologies to approach ROV manoeuvrability have been explored [98, 99]. These methodologies look forward to combining CFD computations and time domain simulation to study ROV manoeuvring features. In [98], a methodology to obtain a simplified hydrodynamic model from CFD computations was implemented. ReFRESCO was chosen as the Reynolds-Averaged Navier-Stokes (RANS) equations solver for CFD computations, and UPB's visor3 ROV was chosen as the study subject. This was the first approach to obtain a manoeuvring model for this ROV. There, hydrodynamic forces were modelled using a tailor-made structure that heuristically matched the observed loads behaviour. CFD calculations were used to compute the corresponding manoeuvring coefficients for different scenarios that considered translation, rotation, and translation-rotation combinations. Conversely, in [99], a methodology for using time-domain simulation software during ROVs' first design stages was tested. MARIN's aNySIM, which specialises on offshore applications, was used as the time-domain simulation software.

1.2.6 CABLE MODELLING

Contrary to the vehicle's model where there is a rather clear path to constructing models (options are not so numerous), modelling a cable (ROV cable) does not have a clear or unique path. In general, the cable behaves as a slender flexible continuum where motion modelling requires joining concepts from kinematics (geometry), dynamics (momentum), elasticity (stress and strain), and numerical methods and discretisation techniques (spatial and temporal). Different approaches can be constructed when different assumptions are stated or choices are made on each of the previous components. Historically, this problem has been theoretically approached assuming the cable is a catenary [64, 1, 66], beam/rod [62, 16, 19], collection of masses [35], or collection of rigid bodies [67, 113, 93].

The catenary approach comes from the classical problem of determining the geometry of a chain that hangs between two points [64]. This approach, commonly used in marine applications [120], assumes that the cable does not have any flexural or torsional strain and may or not have axial strain: the behaviour (geometry and motion) of the cable depends only on the loads. The beam/rod approach assumes, in general, that the cable may exhibit all elasticity phenomena (bending, torsion, and axial strains); compared to the catenary, the behaviour depends on the loads and elastic properties. The collection of masses or lumped-masses is a heuristic approach that often comes from a direct finite elements formulation [35]. The collection of rigid bodies is also a heuristic approach that relies on multibody dynamics models with elasticity constrains obtained from linear elasticity criteria [113, 93].

The catenary is a classical physics/geometry problem. Worth mentioning, Galileo described the geometry of a hanging chain as parabolic, only to be proven wrong half a century later when the Bernoullis, Leibnitz, and Huygens more or less at the same time discovered the catenary geometry [64]. Different approaches may be applied to arrive to a solution (geometry, differential equations, or variational calculus). This idea produced many works and proved highly successful in the ocean engineering community since the 1980's [64, 1, 66].

Regarding beams and rods, the difference between a beam and rod lies in the fact that the shape of the former resembles a straight line and that of the latter a curve [107]. Theories take advantage of the assumption that the longitudinal dimension is long compared to the transversal ones and the body may be considered one-dimensional. This has been a classical subject of elasticity theory [76, 4, 107]. From the elasticity point of view, beam/rod theories could be distinguished by whether they use the Euler-Bernoulli or Timoshenko model of elasticity. Roughly speaking, the former neglects shear strain produced by transversal forces and the latter considers it. The Euler-Bernoulli model is used more often; it considers that the internal moment is proportional to the beam curvature. This follows from the assumption that the transversal area remains normal to the centroids line during deformation and a linear strain-stress constitutive relation. Moreover, the internal transversal (shear) force is derived from a static momentum balance (thus neglecting shear strain). The main assumption may be stated by a constitutive relation where bending moment is proportional to curvature.

In terms of beam/rod kinematics, one approach differs from another depending mostly on the way rotational quantities are encoded and computed. Particularly, rod theories depart from classical linear beam theories because the kinematics of the former are necessarily nonlinear [142], as a consequence of the greater displacements that rods experience. A commonplace alternative is to describe rotations using any Euler angles parametrisation of rotation matrices. In this approach, matrices are used to transform vector quantities between frames, but integrations related to rotation are performed in the space of the Euler angles, for instance, as in early works such as [64, 1, 62]. Worth mentioning, this approach was applied also in the WHOI-cable model: the model used by the Woods Hole Oceanographic Institution [52, 54]. Other approaches describe rotations intrinsically from the curve's derivative, as the case of Nordgren's model [87] applied to underwater slender structures in [49] and ROV cables in [19], where the Frenet-Serret frames are used, and integrations related to rotations are performed in the Euclidean space ($\mathbb{R}^3$). Finally, the seminal work of Simo [114] introduced the so-called *geometrically-exact* approaches, where rotations are directly encoded by rotation matrices and integrated in $SO(3)$, the space (manifold) of rotation matrices. This last approach was used in works such as [16, 97].

Regarding numerical methods for solving catenary and beam/rod models, many approaches have been applied to discretise and integrate the model's differential equations spatially and temporally. Two general techniques for spatial discretisation are finite differences (FD) and the finite element method (FEM). Many early works used FD, *e.g.*, [87, 1, 62, 52, 54]. These models normally use centred differences about the segments' mid-point. FEM has been used in later works [19, 16, 97]. This method offers some advantages such as ensuring continuity and time-space compatibility of the solutions and allowing the formulation of numerical solutions from variational/virtual work principles [114, 16], as well as from differential equations. Time-discretisation and integration often uses classical implicit structural dynamics formulations such as Newmark-β [84] and generalised-α [27] methods. These methods are suited for dynamics problems where the relation among position, velocity, and acceleration is taken into account in the algorithm. The generalised-α method is more general, but Newmark's is more standard. Explicit integration is not common, but *e.g.* in [19] an explicit Runge-Kutta method was used.

Lumped mass and collection of rigid bodies approaches make use of theories from multibody dynamics and, as expected, inherit their different methods. That means that the ways to model and simulate cables are as diverse as that of multibody systems. Often the lumped mass method is favoured because it is cheap to compute, the main reason being a cable is a collection of concentrated masses, so rotational kinetic energy is not taken into account and each mass has three-dimensional dynamics [35]: this is consistent with Euler-Bernoulli beam assumptions. One of the most recognized drawbacks of the method is that when elastic constraints become stiff, most conventional time-integration algorithms become unstable: the resulting discrete system has unstable modes along the axial direction [125].

Models that assume each cable element is a collection of constrained rigid bodies have early developments like the one of Kamman [67], where the dynamic model uses Newton's second law. Later developments use the different methods to solve the motion dynamics via Euler-Lagrange equations: the minimal degrees of freedom and the descriptor approach [138]. The former uses the same idea of robot's dynamics: find a reduced order model where each equation represents a degree of freedom; consequently, internal reaction forces, whose net work is zero, are eliminated. Motion constraints are encoded directly as a function of the degrees of freedom in the definitions of kinetic and potential energy in the Lagrangian. The descriptor form in general includes all six degrees of freedom for each body and encodes the constraints via an augmented Lagrangian, where Lagrange multipliers represent internal impulses that enforce the constraints. The resulting numerical methods become an optimisation algorithm coupled with time integration of the dynamics. Cable dynamics are solved using the reduced order approach by Pereira *et al.* [93] and Gomes *et al.* [55] and the description approach by Servin and Lacoursière [113].

1.2.7 DYNAMIC POSITIONING CAPABILITY

The concept of dynamic positioning capability (DPCap) was originally developed for ships and surface vessels that are capable to operate in dynamic positioning (DP) mode, this is, keeping a position and orientation constant by actively controlling the actuators. More generally, DPCap analyses aim to determine whether craft have enough actuator capacity to withstand environmental and operation-related loads. And furthermore, in a more modern fashion, DPCap evaluates whether a complete vessel's motion system has sufficient capacity to withstand environmental and operational loads while keeping a prescribed motion at normal and failure conditions.

In modern autonomous or intelligent water craft, the dynamic positioning (DP) system is responsible for distributing power to the actuators in order to keep a constant position and heading or attain a prescribed motion. This idea of DP started to be applied in surface vessels in the 1960s, with a significant increase since the late 1970s mainly thanks to advances in multivariable control and Kalman filtering theories. A first formal description of a ship DP system was made by Balchen [11]. The DP system usually includes, as subsystems, power generation, control system (computer, sensors, position reference systems, operator panels, etc.), actuator system, and power management. Consequently, a DP system is able to automatically balance the actuators' forces and moments with the environmental and operation-specific loads and any other disturbances, often by keeping precise position and orientation control while maintaining low power consumption [121].

DPCap is usually illustrated by using DPCap plots. DPCap plots are polar graphs that show the operation envelope of a vessel by depicting the limiting environmental conditions it can withstand in DP mode, at each direction (or at least a set of directions), and at normal and failure conditions. An early reference on the use of these plots can be found in the work by Webb [140], where the DP and operability capabilities of the Ocean Clipper drillship were studied. Later, the International Marine Contractors Association (IMCA) released the “Specification for DP Capability Plots (IMCA M 140)” [63], which standardised the requirements to elaborate and present DP capability plots. There, a quasi-static method to balance environmental loads and actuators forces was proposed. In this context, a quasi-static balance is a steady-state balance where loads are augmented by a factor of safety that accounts for unsteady phenomena and uncertainties.

Following the release of IMCA M 140 standard, DPCap analyses started to be used for vessel design processes [82, 132], contractor decision making [12], reliability assessment [102], among others. For instance, Mahfouz and El-Tahan [82] developed software for creating DPCap plots of marine vessels and floating production units for Oil and Gas industries, this software was declared by the authors as a potential tool for thruster selection and configuration. This work was further expanded by Mahfouz [81] where neural networks where used to enhance the standard DPCap plots predictions. Some works propose DPCap as a tool that may be used during vessel early design stages, such as those of van ’t Veer and Gachet [132], Luebke *et al.* [79] and Antheunisse *et al.* [3]. Other works proposed DPCap as detailed design/optimisation tool, such as thruster sensitivity analyses to assess how each individual thruster influences the overall capability [144], as well as local optimisation of the thruster configuration [143].

Important drawbacks of the IMCA M 140’s quasi-static method were stated later by different authors, including the necessity to develop methods in order to include dynamic effects and operation-specific environmental and external loads. For instance, Serraris [112] showed the use of time-domain simulation software to analyse the DP capabilities of a mono-hull deep-water drillship including a Kalman filter, PID controller, and thruster interaction effects. Later, works by Smogeli *et al.* [119] and Nguyen *et al.* [85] proposed a formal expansion of the DP capability methodologies to the dynamic case, including descriptions of the most important dynamics to consider. Moreover, among the studies where operation- and site-specific conditions were considered, one can find works were ice operation [68], depth-related effects [18], and multi-body operations [51] were analysed.

More recently, the DNVGL-ST-0111 standard was released [34]. This standard gives indications on how to calculate and document DP capability analysis for surface vessels in quasi-static and dynamic conditions, when considering either standard and site-specific environmental conditions and loads. It also gives a prescriptive method to make comparisons among mono-hulled ship-shaped vessels. The standard gives indications on how to calculate and report the DP capability of a vessel using DP capability numbers. According to the standard "The DP capability number indicates that a vessel's position keeping ability can be maintained in the corresponding DP capability number condition and all conditions below, but not in the conditions specified for the next DP capability number." These numbers are given using the Beaufort Scale as parallel and related to certain wind, waves, and current conditions. The standard comprises three levels, but a total of five types of analyses: Level 1, Level 2, Level 2-Site, Level 3, and Level 3-Site. For DP capability Level 1, Level 2 and Level 2-Site the calculations are made considering quasi-static conditions: the main idea is to start balancing the lowest environmental conditions and increase the weather conditions until a limiting condition is reached, according to the DP capability number definition. For the DP capability Level 3 and Level 3-Site dynamic conditions are considered. Level 1 corresponds to a prescriptive method for analysing ship-shaped mono-hull vessels. Levels 2 and 3 are for any surface vessel shape and allow the inclusion of model specifics. The "Site" variation of Levels 2 and 3 allow the inclusion of site/operation specific environmental conditions and loads.

As part of the current thesis, in a work by Ramírez-Macías *et al.* [101], a methodology for DP capability studies for ROVs based on the DNVGL-ST-0111 standard [34] was proposed. There, ROV and submerged craft specifics and differences to surface vessels were taken into account. More specifically, ocean current was considered the primary environmental variable to determine capability, 2D and 3D plots where proposed to illustrate three-dimensional motion capability and account for three-dimensional relative current, and the effects of vertical current profile on the cable and other depth- and operation-related were included. Potentially, DPCap studies on ROVs can be used as a design tool for developers and as a decision making tool for ROV buyers, contractors, and operators; for instance, to allow customers and contractors know whether an ROV was appropriately designed for a specific operation and choose the appropriate one for their needs. It also lets the ROV operators to plan the operations using environmental data and ROV-specific information and implement on-line operation assessment. And lastly, gives ROV designers a framework to assess, evaluate, and optimise the DP system components.

1.2.8 MOTION FEASIBILITY FRAMEWORK

Based on the reviewed literature and to the author's knowledge, a methodology to study motion feasibility that integrates the analysis of many motion-related subsystems is not available, as it is proposed in this work. Such methodology is grounded on the aforementioned theory to model the ROV and cable, design of feedback control, and thrust allocation. Moreover, given limitations to the way hydrodynamic studies are currently performed, this thesis is intended to propose means to study the hydrodynamics of an ROV by using global, simplified models assembled from data obtained through CFD computations. This work aims at closing the knowledge gap in assessing the motion capabilities of an ROV system integrally.

1.3 PROBLEM STATEMENT

Motivated by the idea of developing better remotely controlled underwater systems, this work proposes the development of analytical- and numerical-oriented tools for the dynamic analysis and motion control of ROVs, using an integrated approach where hydrodynamics, control, and cable effects are considered and aimed at answering questions about motion feasibility. Such analytical and numerical tools are expected to facilitate the analysis, design, and operation of systems that involve ROVs; for example, either planning an exploration mission or analysing, designing, or improving an exploration system. These tools are intended to be useful to any ROV system, but in this work they are applied to and illustrated on observation and medium-size ROVs of which information is currently available.

The approach proposed in this work encompasses the following tasks: at first, it includes studying, testing, and proposing suitable simulation models, using existing ROV motion modelling and control theory, but integrating more elaborate hydrodynamic models (compared to those in, *e.g.*, [48, 36]), as well as effects of cable mechanics; second, there should be an implementation of numerical computations on the models, including testing different basic simulation scenarios; and, third, an implementation of an integrated framework for analysing and designing systems and operational schemes, where components such as hydrodynamics, control, and cable are considered. This work will be grounded on available theory to model the ROV and cable, design of feedback control, and thrust allocation; furthermore, it looks forward to contributing means to study the hydrodynamics of an ROV and developing a framework where hydrodynamics and other motion-relevant systems can be integrated and evaluated. In a nutshell, this work requires understanding and implementing available models as a baseline to propose an integrated framework to analyse multiple components that constrain an ROV system's motion.

For the hydrodynamics part, studying the manoeuvring behaviour of ROVs using time-domain simulation and computational fluid dynamics (CFD) is proposed. Here, a CFD viscous-flow solver can be used by applying the virtual captive test (VCT) approach, where experimental fluid dynamics conditions are simulated using numerical computations. From the computations, a simplified manoeuvring model, suitable for fast time-domain simulation (compared to CFD), can be assembled and further implemented in other analyses. The main question to solve in this context is how to use CFD practically to obtain a meta-model useful for time-domain simulation; this requires proposing a methodology to plan the computations and take advantage of data coming from them. As part of this, different ways of performing VCT, namely CFD computations, as well as assembling a manoeuvring meta-model will be studied and analysed. Given the computational nature of this work, how to quantify numerical model uncertainty, namely solution verification, shall also be studied.

For the cable model part, given the fact that the dynamics of beams, elastic rods, and other slender structures have been widely studied, there are numerous approaches when, for instance, different kinematics, dynamics, elasticity, and time and space discretisation schemes are to be considered. The approaches' spectrum range from relatively simple and heuristic to highly convoluted and rigorous. This work will cope with the challenge of studying and evaluating different approaches and integrating at least one cable model into the analysis framework. The main question to solve in this context is what model or models are suitable for solving motion feasibility analyses practically. Because the model spectrum is wide, the idea is to test catenary models first and assess their utility, and further on study if increasing the model's complexity to include elasticity are feasible and practical.

As a first approach to integrating concepts from motion modelling and control, dynamic positioning capability (DPCap) studies are proposed to solve design- and operational-related questions. The concept of DPCap was first introduced for surface vessels that include a dynamic positioning (DP) system, in order to assess whether they are able to withstand environmental and operational loads while keeping a desired position or path. In this work, DPCap studies will be extended from surface vessels to ROVs, considering features that make ROVs different, such as three-dimensional motion and the presence of a cable. The concept of DPCap is useful either for offline studies such as ROV motion system design, control tuning, and operation planning, as well as for operation assistance as an online advisory tool that assess in real time whether the system is able to operate at certain conditions.

By using this integrated approach, this work will address the question about the motion feasibility of an ROV system. This is equivalent to ask, given some environmental conditions, *will the ROV be able to operate?* This question has a trivial answer: if the water is calm (lack of waves and current), yes, but if the water is not calm, maybe. This is valid for any stable, neutrally buoyant ROV system because breaking equilibrium in water is always possible. Furthermore, providing any other non-trivial answer is difficult. This fact makes the question interesting, because there is no such thing as motionless water in practical situations. Asking whether the system will be able to operate or not at certain conditions requires proposing a way to quantify the level of motion capability, in order to compare among different systems or configurations for the same system. This work intends to be a first approach to ask about motion feasibility joining vehicle dynamics, hydrodynamics, cable mechanics, and control.

1.4 OBJECTIVES

1.4.1 GENERAL

This thesis is oriented to the accomplishment of one general objective: *To develop a framework for dynamics and motion control studies on remotely operated vehicles (ROV) and highly flexible elastic rods, oriented to the analysis and design of ROVs and their operations and aimed at answering questions about motion feasibility.* This general objective is carried through four specific objectives. In the next section, the reader can track what chapter or chapters of the book address each specific objective.

1.4.2 SPECIFIC

To elaborate simplified ROV hydrodynamics models to be used in time-domain simulation and control design by means of viscous-flow computations and the virtual captive test approach, is addressed in Chapter 6, Hydrodynamics.

To mathematically model highly flexible elastic rods that are used in ROV operations, including the integration into a framework for ROV motion feasibility studies, is addressed in Chapter 7 principally and in Chapter 10 rather tangentially.

To implement time-domain simulations of ROVs with highly flexible elastic rods, where manoeuvring and control-scheme concepts are tested and evaluated, is addressed in Chapters 8, 9, and 10.

To develop a DP capability analysis framework for ROV system motion feasibility analyses, where steady-state and unsteady motion scenarios are included, is addressed in Chapter 9 and, more specifically, Chapter 10.

1.5 CONTRIBUTION AND DELIVERABLES

1.5.1 CONTRIBUTION

This work is mainly about integrating the analysis of many motion components that are traditionally considered separately, in order to study motion feasibility. It is is grounded on available theory to model the ROV and cable, design of feedback control, and thrust allocation. Conversely, it is expected to contribute means to study the hydrodynamics of an ROV and develop a framework where hydrodynamics and other motion-relevant systems can be integrated and evaluated. This work pursues the advance in knowledge about assessing the motion capabilities of an ROV system. This is done through the following contributions:

- A procedure to obtain ROV manoeuvring models from viscous flow computations.
- A framework to analyse an ROV system's motion feasibility integrally, including ROV, cable, feedback control, and thrust allocation, by means of ROV dynamic positioning capability analysis.

1.5.2 PRODUCTS

During the author's PhD studies, different products were accomplished

- Thesis monograph.
- Published articles:
  - Hydrodynamic modelling for the remotely operated vehicle VISOR3 using CFD [98]. Published in IFAC-PapersOnLine.
  - A methodology for DP capability studies on remotely operated vehicles [101]. Published in Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering—OMAE.
  - On the use of time-domain simulation in the design of Remotely Operated Vehicles [99]. Published in Ship Science & Technology.
- Further articles:
  - Article on DP Cap.
  - Article on ROV hydrodynamics and manoeuvring.
- Software suite in Python covering the following topics:
  - DP Cap.
  - ROV time-domain simulation.

- Cable.
- Environment modelling.

1.6 THESIS ORGANISATION

This thesis brings together ideas from rigid body dynamics, control theory, time-domain simulation, object oriented programming and related scientific computing software, some spherical geometry, design of (computer) experiments, hydrodynamics and use of CFD tools, and structural mechanics and its application to slender bodies. With these tools in hand, the thesis proposes means to study and analyse an ROV system's motion feasibility, *i.e.*, whether operating at some known conditions is possible or not. The theme is in nature for someone who has background in classical mechanics, control, structural mechanics/elasticity, hydrodynamics, and programming, and feels fine reading maths. Acknowledging that in general it is difficult to find someone that feels comfortable around all the aforementioned topics, this book is written to cover the basic tools for understanding the whole matter. Because of this and since the thesis covers many topics, the text is rather long; consequently, the author appeals to the reader's patience. The description of the thesis organisation herein looks forward to inform the reader about what to find where and allow each person to adjust to their own interests.

The book is divided into four parts that succeed this introductory chapter: (I) Basics, (II) Hydrodynamics and cable modelling, (III) Manoeuvring and motion feasibility, and an (IV) Epilogue. Part I, *Basics*, comprises the building blocks of the forthcoming work. The chapters contained in this part do not claim any contribution besides the way they are presented; nevertheless, they are included in order to make the book complete. Part I contains four chapters: from Chapter 2 to 5. Chapter 2, *Vehicle motion modelling and control*, deals with usual rigid body dynamics and control for water craft but particularised for ROVs. Chapter 3, *Study subjects*, illustrates the particularities of the ROVs to be used further in the thesis, with an emphasis on the information required for modelling their dynamics. Chapter 4, *Tools from sphere geometry*, lays some tools from sphere geometry that ease dealing with problems that explore and interpret spaces with a spherical geometry. Finally, Chapter 5, *Design of computer experiments*, presents a collection of tools to organise series of computations and assess them by using metrics.

Part II, *Hydrodynamics and cable modelling*, is comprised by Chapters 6 and 7. These two chapters deal with two noteworthy parts of an ROV model that affect motion significantly and are cumbersome to model: hydrodynamics and cable. Chapter 6, *Hydrodynamics*, contributes means to model ROV hydrodynamics. This chapter presents a methodology for obtaining a meta-model that is suited for time-domain simulation and whose base information comes from viscous-flow computations. This is carried through by describing the methodology and applying the complete process to an ROV (VISOR3). Conversely, Chapter 7, *Cable modelling*, contributes means to model the cable. The chapter builds knowledge in modelling and performing computations on ROV cables by starting with the classical static catenary and, from that point on, increasing the model's complexity. In brief, the chapter approaches computations on the catenary by using different numerical methods (shooting algorithm, finite differences, and finite elements) and then studies the effect of elasticity by solving the WHOI cable model (using finite differences).

Part III, *Manoeuvring and motion feasibility*, is comprised by Chapters 8, 9, and 10. Chapter 8, *Time-domain simulation framework*, shows a framework that uses object-oriented programming for performing time-domain simulations on ROVs. This framework is used to assemble the time-domain simulations computed throughout the thesis. Chapter 9, *Manoeuvring time-domain simulations*, could be understood a follow-up chapter to Chapter 6, *Hydrodynamics*, that makes use of the time-domain framework presented in Chapter 8. The chapter comprises a number of time-domain manoeuvring computations that use VISOR3's model to study qualitatively and quantitatively its manoeuvrability. Chapter 10, *Motion feasibility framework*, presents the concept of ROV dynamic positioning capability (ROV-DPCap) as the means to perform motion feasibility analyses. This chapter wraps-up the main purpose of the thesis: to assess whether motion of an ROV is feasible under foreknown conditions, by means of numerical computation of the time-domain dynamics equations. Following this last chapter, the thesis closes with Part IV, an *Epilogue.* This part includes commonplace Conclusions and Further Work, References, and Appendices.

Part I

Basics

Chapter 2

Vehicle motion modelling and control

This work relies on being able to simulate ROV motion in the time domain. The ROV simulation model is based on rigid-body Newtonian mechanics, where external loads are assumed as Euclidean vector quantities. The approach presented herein is mainly based on Fossen's theory [48, 36] and is complemented by some ideas and notation coming from the field of geometric mechanics (see for instance [15, 88]). All models presented in this chapter are grounded on existing theory and represent the baseline vehicle model to be further used.

2.1 KINEMATICS

Roughly speaking, *kinematics* deals with how to define and manipulate position, orientation, and velocity quantities of moving bodies such as water craft. Kinematics is geometrical in nature. Position and orientation are also called *configuration* or *pose*, and all possible configurations define the *configuration space* [72, 15, 88]. When the motion of a rigid body does not cover great distances, it is enough to define its configuration by using a North-East-Down (NED) frame $\{n\}$ and a body-fixed frame $\{b\}$ [48], see Fig. 2.1. Frame $\{n\}$ is assumed inertial and defined on a tangent plane on the Earth's surface. The frame is orthogonal and directions x-y-z, as its name suggests, point towards North-East-Down respectively. The origin may be placed at any feasible coordinates, but commonly z is zero at the water surface.

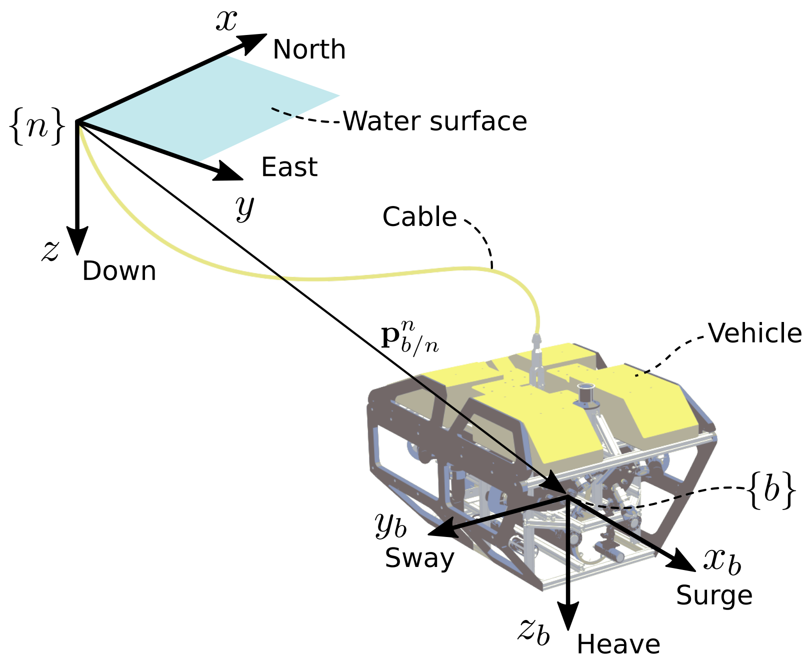


Figure 2.1. Frames to describe ROV motion. This figure was created by the author from a render of Pionero500's CAD model. The CAD model was the work of many of the UPB's Underwater Robotics Program students and researchers. The render was finally obtained thanks to Sergio Suárez.

2.1.1 POSITION AND ORIENTATION

The *position* of the body is given by $\mathbf{p}^n_{b/n} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T} \in \mathbb{R}^3$, as the position of the origin of $\{b\}$ with respect to $\{n\}$. Conversely, the orientation is described by the rotation matrix between $\{n\}$ and $\{b\}$, $\mathbf{R}^n_b \in \mathbb{R}^{3\times3}$, that maps an arbitrary vector $\mathbf{v}^b \in \mathbb{R}^3$ defined in $\{b\}$ to a vector $\mathbf{v}^n$ defined in $\{n\}$, *i.e.*, $\mathbf{v}^n = \mathbf{R}^n_b\mathbf{v}^b$. More specifically, admissible matrices are elements of $SO(3)$, the special orthogonal group of order 3, *i.e.*, $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3\times3} \mid \mathbf{R}\,\text{orthogonal}, \quad \det\mathbf{R} = 1\}$; $SO(3)$ is a three-dimensional subspace of $\mathbb{R}^{3\times3}$ (in differential geometry, such subspace is a manifold).

To define *orientation*, a three-dimensional parametrisation based on the roll-pitch-yaw Euler angles ϕ, θ, and ψ may be used. This parametrisation is obtained by successive rotations using the $z{-}y{-}x$ convention. The Euler angles vector, often called *attitude*, is $\mathbf{\Theta}_{nb} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\mathsf{T}$. This parametrisation maps a subset of $\mathbb{R}^3$ to $\mathbb{R}^{3\times3}$. Consequently, $\mathbf{R}^n_b$, where $\mathrm{s}\cdot = \sin(\cdot)$ and

$\mathrm{c}\cdot = \cos(\cdot)$, is

$$\mathbf{R}_b^n\left(\mathbf{\Theta}_{nb}\right) = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{c}\phi\mathrm{s}\theta \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\theta\mathrm{s}\psi\mathrm{c}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix}.$$

Hence, the *configuration* may be described by $\eta \in \mathbb{R}^6$,

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{b/n}^n \\ \mathbf{\Theta}_{nb} \end{bmatrix}.$$

Other configuration descriptions are possible depending on how the orientation is represented. For instance, using quaternions or the rotation matrix directly [48]. The latter is used to develop numerical integration algorithms often called *Lie group integrators* [72, 88].

2.1.2 VELOCITY

The body's *linear velocity* is given on $\{b\}$ by $\mathbf{v}_{b/n}^b \in \mathbb{R}^3$, $\mathbf{v}_{b/n}^b = \begin{bmatrix} u & v & w \end{bmatrix}^\mathsf{T}$, and the *angular velocity* by $\omega_{b/n} \in \mathbb{R}^3$, $\omega_{b/n} = \begin{bmatrix} p & q & r \end{bmatrix}^\mathsf{T}$. The angular velocity is often expressed as $\mathbf{\Omega}_{b/n} \in \mathfrak{so}(3)$, where $\mathfrak{so}(3) \subset \mathbb{R}^{3\times3}$ is the set of skew-symmetric matrices, *i.e.*, $\mathfrak{so}(3) = \{\mathbf{\Omega} \in \mathbb{R}^{3\times3} |\, \mathbf{\Omega} = -\mathbf{\Omega}^\mathsf{T}\}$. The isomorphism $\mathrm{S} : \mathbb{R}^3 \to \mathfrak{so}(3)$ is often defined as

$$\mathbf{\Omega}_{b/n} = \mathrm{S}(\omega_{b/n}) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}.$$

The *velocity vector* is thus given by $\nu \in \mathbb{R}^6$,

$$\nu = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \omega_{b/n} \end{bmatrix}. \tag{2.1}$$

The set (Cartesian product) of a configuration and velocity is called the vehicle's *state*. Using the aforementioned definitions, the state $\mathbf{q} \in \mathbb{R}^{12}$ is given by $\mathbf{q} = \begin{bmatrix} \eta^\mathsf{T} & \nu^\mathsf{T} \end{bmatrix}^\mathsf{T}$.

2.1.3 KINEMATIC TRANSFORMATION

The *kinematic transformation* is used to define the tangent space of a specified configuration η. In rigid-body kinematics it is commonly expressed as a mapping defined by a $\mathbf{J}(\eta) \in \mathbb{R}^{6\times6}$. In general terms, this mapping is defined as

$$\dot{\eta} = \mathbf{J}(\eta)\nu, \tag{2.2}$$

where the overdot indicates differentiation with respect to time. Given a fixed configuration η, $\mathbf{J}(\eta)$ linearly maps ν into $\dot{\eta}$.

For instance, when attitude vector $\boldsymbol{\Theta}_{nb}$ encodes the orientation, the mapping is [48]

$$\mathbf{J}_{\Theta}(\eta) = \begin{bmatrix} \mathbf{R}_b^n\left(\boldsymbol{\Theta}_{nb}\right) & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{T}_{\Theta}\left(\boldsymbol{\Theta}_{nb}\right) \end{bmatrix}, \tag{2.3}$$

where

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

$\mathrm{s}\cdot = \sin(\cdot)$, $\mathrm{c}\cdot = \cos(\cdot)$, $\mathrm{t}\cdot = \tan(\cdot)$, and $\mathbf{0}_{3\times 3} \in \mathbb{R}^{3\times 3}$ is a zeros matrix. The main drawback of this mapping is that it becomes singular when $\theta = \pm\pi/2$, this fact motivates the use of quaternions when this singular configuration may be reached [48]; this is hardly the case of ROVs. The kinematic transformation (2.2) using (2.3) is then defined as

$$\dot{\eta} = \mathbf{J}_{\Theta}(\eta)\nu.$$

2.1.4 RELATIVE VELOCITY KINEMATICS AND FLOW FRAME

The dynamics model requires a definition of the body's relative velocity, *e.g.*, when computing hydrodynamic loads. In this work, *relative velocity* is defined with respect to the current around the vehicle. In general, (ocean) *current* is a function of time and space, *i.e.*, $\mathbf{V}_{c/n}^n = \mathbf{V}_{c/n}^n(x, y, z, t)$. Once an operation site is known, it is common to assume that current varies with depth and time $\mathbf{V}_{c/n}^n = \mathbf{V}_{c/n}^n(z, t)$. This current profile may be known through field measurements or ocean models. Nevertheless, when such models are not available, the DNV-RP-C205 Environmental Conditions and Environmental Loads Recommended Practice presents a simplified model for the steady-state case [33].

This work assumes that currents are irrotational and their acceleration is negligible. This means that for the irrotational case, current with respect to $\{n\}$ is

$$\mathbf{V}_{c/n}^n = \begin{bmatrix} v_{c_x} & v_{c_y} & v_{c_z} & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}.$$

Then, by using the inverse of (2.3), current velocity with respect to $\{b\}$ is

$$\nu_c = \mathbf{J}_{\Theta}(\eta)^{-1}\mathbf{V}_{c/n}^n;$$

subsequently, *relative velocity* with respect to $\{b\}$ is given by

$$\nu_r = \nu - \nu_c$$

and the relative acceleration by

$$\dot{\nu}_r = \dot{\nu}.$$

The last expression facilitates the computation of the dynamics by assuming that the vehicle's acceleration equals the relative acceleration.

A convenient way to represent relative velocity is presented in [48], using the so-called FLOW frame. This frame is written using the drift/side-slip and attack angles: β and α respectively. Using this description, relative velocity is given by

$$\nu_r = \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} V_r \cos(\alpha_r)\cos(\beta_r) \\ V_r \sin(\beta_r) \\ V_r \sin(\alpha_r)\cos(\beta_r) \end{bmatrix}, \tag{2.4}$$

where V_r is relative velocity's magnitude, α_r attack angle, and β_r drift/sideslip angle. In this work, this representation has two practical uses: parametrising relative currents on three-dimensional computations directly, as well as exploratory analysis and visualisation.

2.2 RIGID BODY DYNAMICS

The dynamic model of an ROV can be obtained from the Newton-Euler equations of motion when frame $\{b\}$'s origin is located at a generic, geometrically convenient point (denoted as CO); in general, this point does not coincide with the centre of mass. The dynamic model could be derived also from a variational principle such as the Lagrange-d'Alembert principle [88]. Nevertheless, in this work Fossen's robot-like equation derived from the Newton-Euler formulation is preferred. This equation is given in vector form ($\in \mathbb{R}^6$) by [48]

$$\mathbf{M}_{RB}\dot{\nu} + \mathbf{C}_{RB}\left(\nu\right)\nu + \mathbf{M}_A\dot{\nu}_r + \mathbf{C}_A\left(\nu_r\right)\nu_r + \mathbf{D}\left(\nu_r\right)\nu_r + \mathbf{g}\left(\eta\right) = \tau(\mathbf{u}) + \tau_{\text{cable}} + \tau_{\text{wave}}. \tag{2.5}$$

This model includes loads from rigid body kinetics (inertia and Coriolis), hydrodynamics (drag, added mass, and hydrodynamic Coriolis), hydrostatics, actuators, and other external forces, *e.g.*, cable or wave loads.

More specifically, $\mathbf{M}_{RB}$ and $\mathbf{C}_{RB} \in \mathbb{R}^{6\times6}$ represent rigid-body mass and Coriolis matrices; $\mathbf{M}_A$ and $\mathbf{C}_A \in \mathbb{R}^{6\times6}$ represent hydrodynamic added mass and Coriolis matrices; $\mathbf{D} \in \mathbb{R}^{6\times6}$ represents hydrodynamic drag; $\mathbf{g} \in \mathbb{R}^{6\times1}$ represents hydrostatic forces (buoyancy and restoring moments); $\tau \in \mathbb{R}^{6\times1}$ is the force vector due to actuators, dependent on some input $\mathbf{u}$; $\tau_{\text{cable}} \in \mathbb{R}^{6\times1}$ is the force vector due to the cable; and $\tau_{\text{wave}} \in \mathrm{R}^{6\times1}$ is the force vector due to waves. Nonetheless, in the case of waves, the effects, divided into first and second order loads, can be disregarded when the ROV is submerged more than half of the wave length [120].

2.2.1 RIGID BODY INERTIA

The rigid body inertial model is obtained when one knows the vehicle's mass m, as well as the tensor of inertia $\mathbf{I}_b$ and location of the centre of mass $\mathbf{r}_g^b = \begin{bmatrix} x_g & y_g & z_g \end{bmatrix}^\intercal$, both with respect to CO. Then, the matrix of inertia is given by

$$\mathbf{M}_{RB} = \left[\begin{array}{c|c} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \hline \mathbf{M}_{21} & \mathbf{M}_{22} \end{array}\right], \tag{2.6}$$

where $\mathbf{M}_{11} = m\mathbf{I}_{3\times3}$ is the mass matrix,

$$\mathbf{M}_{22} = \mathbf{I}_b = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix}$$

is the inertia tensor measured with respect to CO, and

$$\mathbf{M}_{21} = \mathbf{M}_{12}^\intercal = m\mathrm{S}(\mathbf{r}_g^b) = \begin{bmatrix} 0 & -mz_g & my_g \\ mz_g & 0 & -mx_g \\ -my_g & mx_g & 0 \end{bmatrix}.$$

Once $\mathbf{M}_{RB}$ is known, the rigid-body Coriolis matrix is

$$\mathbf{C}_{RB} = \begin{bmatrix} \mathbf{0}_{3\times3} & -\mathrm{S}\left(\mathbf{M}_{11}\nu_1 + \mathbf{M}_{12}\nu_2\right) \\ -\mathrm{S}\left(\mathbf{M}_{11}\nu_1 + \mathbf{M}_{12}\nu_2\right) & -\mathrm{S}\left(\mathbf{M}_{21}\nu_1 + \mathbf{M}_{22}\nu_2\right) \end{bmatrix}. \tag{2.7}$$

Furthermore, when a starboard-port-symmetrical ROV is assumed, (2.6) can be simplified [48]. This means that $y_g = 0$ and $I_{xy} = I_{yz} = 0$. In this case

$$\mathbf{M}_{22} = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}$$

and

$$\mathbf{M}_{21} = \mathbf{M}_{12}^\intercal = \begin{bmatrix} 0 & -mz_g & 0 \\ mz_g & 0 & -mx_g \\ 0 & mx_g & 0 \end{bmatrix}.$$

Hence, the rigid body inertia becomes

$$\mathbf{M}_{RB} = \left[\begin{array}{ccc|ccc} m & 0 & 0 & 0 & mz_g & 0 \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & 0 & -mx_g & 0 \\ \hline 0 & -mz_g & 0 & I_x & 0 & -I_{xz} \\ mz_g & 0 & -mx_g & 0 & I_y & 0 \\ 0 & mx_g & 0 & -I_{xz} & 0 & I_z \end{array}\right].$$

2.2.2 HYDROSTATICS

The hydrostatics model is obtained when one knows the vehicle's weight W and buoyancy B, as well as the location of the centre of buoyancy $\mathbf{r}_b^b = \begin{bmatrix} x_b & y_b & z_b \end{bmatrix}^\mathsf{T}$ with respect to CO. Then, the hydrostatics load becomes [48]

$$\mathbf{g}(\eta) = \begin{bmatrix} (W-B)\sin\theta \\ -(W-B)\cos\theta\sin\phi \\ -(W-B)\cos\theta\cos\phi \\ -(y_gW-y_bB)\cos\theta\cos\phi-(z_gW-z_bB)\cos\theta\sin\phi \\ (z_gW-z_bB)\sin\theta+(x_gW-x_bB)\cos\theta\cos\phi \\ -(x_gW-x_bB)\cos\theta\sin\phi-(y_gW-y_bB)\sin\theta \end{bmatrix}.$$

In this equation, weight may be calculated as

$$W = (m+m_p)g, \tag{2.8}$$

where m_p is payload mass, and buoyancy is given by

$$B = \rho_\mathrm{W} g \nabla,$$

where ∇ is displaced volume and ρ_W water's density.

Assuming xz symmetry and that the vehicle is well balanced (*i.e.*, $x_gW = x_bB$ and $y_gW = y_bB$), this becomes [48]

$$\mathbf{g}(\eta) = \begin{bmatrix} (W-B)\sin\theta \\ -(W-B)\cos\theta\sin\phi \\ -(W-B)\cos\theta\cos\phi \\ -(z_gW-z_bB)\cos\theta\sin\phi \\ (z_gW-z_bB)\sin\theta \\ 0 \end{bmatrix}.$$

This could be further simplified if one assumes that the vehicle is neutrally buoyant, meaning that $W = B$. In this case,

$$\mathbf{g}(\eta) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -W(z_g-z_b)\cos\theta\sin\phi \\ W(z_g-z_b)\sin\theta \\ 0 \end{bmatrix}.$$

In this last case, hydrostatics only includes roll and pitch restoring moments.

2.2.3 HYDRODYNAMICS

As shown in (2.5), hydrodynamics terms include added mass, hydrodynamic Coriolis, and drag, *i.e.*, $\mathbf{M}_A\dot{\nu}_r + \mathbf{C}_A\left(\nu_r\right)\nu_r + \mathbf{D}\left(\nu_r\right)\nu_r$. These terms include hydrodynamic effects dependent on relative acceleration, linear and angular relative velocities, and linear-angular cross-coupled effects.

2.2.3.1 Added mass and hydrodynamic Coriolis

Added mass, that includes effects dependent on relative acceleration, is given in general by

$$\mathbf{M}_A = -\begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{p}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{p}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{p}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{p}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{p}} & N_{\dot{r}} \end{bmatrix}. \tag{2.9}$$

This matrix is assumed symmetric and positive definite, *i.e.*, $\mathbf{M}_A = \mathbf{M}_A^{\intercal} \geq 0$.

The hydrodynamic Coriolis matrix, that includes linear-angular cross-coupled effects, could be obtained from $\mathbf{M}_A$ by following a similar procedure as in (2.7). This means that

$$\mathbf{C}_A = \begin{bmatrix} \mathbf{0}_{3\times3} & -\mathrm{S}\left(\mathbf{A}_{11}\nu_1 + \mathbf{A}_{12}\nu_2\right) \\ -\mathrm{S}\left(\mathbf{A}_{11}\nu_1 + \mathbf{A}_{12}\nu_2\right) & -\mathrm{S}\left(\mathbf{A}_{21}\nu_1 + \mathbf{A}_{22}\nu_2\right) \end{bmatrix}. \tag{2.10}$$

When operating below the wave-affected zone, where wave frequency may be assumed equal to zero, hydrodynamic added mass is constant and for ROV-like bodies may be simplified to [48]

$$\mathbf{M}_A = -\begin{bmatrix} X_{\dot{u}} & 0 & X_{\dot{w}} & 0 & X_{\dot{q}} & 0 \\ 0 & Y_{\dot{v}} & 0 & Y_{\dot{p}} & 0 & Y_{\dot{r}} \\ X_{\dot{w}} & 0 & Z_{\dot{w}} & 0 & Z_{\dot{q}} & 0 \\ 0 & Y_{\dot{p}} & 0 & K_{\dot{p}} & 0 & K_{\dot{r}} \\ X_{\dot{q}} & 0 & Z_{\dot{q}} & 0 & M_{\dot{q}} & 0 \\ 0 & Y_{\dot{r}} & 0 & K_{\dot{r}} & 0 & N_{\dot{r}} \end{bmatrix}. \tag{2.11}$$

Furthermore, assuming low speed and three planes of symmetry, the contribution from the off-diagonal elements could be neglected [48]. This means that

$$\mathbf{M}_A = -\operatorname{diag}\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\}.$$

2.2.3.2 Linear and quadratic drag

A simplified way to compute $\mathbf{D}\left(\nu_r\right)$ is by assuming that forces are uncoupled and composed only of combined linear and quadratic terms. This means that [48]

$$\begin{aligned}\mathbf{D}(\nu_r) = &-\operatorname{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\}\\ &-\operatorname{diag}\{X_{|u|u}|u|, Y_{|v|v}|v|, Z_{|w|w}|w|, K_{|p|p}|p|, M_{|q|q}|q|, N_{|r|r}|r|\}.\end{aligned} \tag{2.12}$$

Moreover, it may be considered that linear damping is faded out with increasing speed [120].

2.2.3.3 Manoeuvring meta-model

In (2.5), all hydrodynamics terms may be summarised as

$$\tau_H\left(\nu_r, \dot{\nu_r}\right) = -\left(\mathbf{M}_A\dot{\nu_r} + \mathbf{C}_A\left(\nu_r\right)\nu_r + \mathbf{D}\left(\nu_r\right)\nu_r\right); \tag{2.13}$$

consequently, the dynamic model becomes

$$\mathbf{M}_{RB}\dot{\nu} + \mathbf{C}_{RB}\left(\nu\right)\nu + \mathbf{g}\left(\eta\right) = \tau_H\left(\nu_r, \dot{\nu_r}\right) + \tau(\mathbf{u}) + \tau_{\text{cable}} + \tau_{\text{wave}}.$$

Thus, hydrodynamic loads in this context may be understood as a load term that is dependent on relative velocity ν_r and acceleration $\dot{\nu_r}$ and modelled as an algebraic expression. It is often useful to write τ_H's components explicitly, *i.e.*,

$$\tau_H = \begin{bmatrix} X & Y & Z & K & M & N \end{bmatrix}^{\mathsf{T}},$$

where X, Y, and Z are x-y-z forces respectively and K, M, and N are x-y-z moments respectively. This is called the *component form.*

Using the concept of hydrodynamic derivatives [48], where τ_H is written as a Taylor-series expansion, hydrodynamic loads are modelled by polynomial expressions. Coefficients of resultant polynomial terms are *hydrodynamic derivatives.* For example, one can write for X the linear expansion

$$X = X_u u + X_v v + X_w w + X_p p + X_q q + X_r r + X_{\dot{u}}\dot{u} + X_{\dot{v}}\dot{v} + X_{\dot{w}}\dot{w} + X_{\dot{p}}\dot{p} + X_{\dot{q}}\dot{q} + X_{\dot{r}}\dot{r},$$

where coefficients $X_{(\cdot)}$ are hydrodynamic derivatives. Furthermore, higher order expansions may be used as well. In this case, often added-mass terms are linear and drag terms use only odd elements, given drag's dissipative nature. Hence, for X one obtains

$$X = X_{\dot{u}}\dot{u} + X_u u + X_{uuu}u^3 + X_{\dot{v}}\dot{v} + X_v v + X_{vvv}v^3 + X_{\dot{w}}\dot{w} + X_w w + X_{www}w^3 + \cdots.$$

Moreover, second order terms may be included in modulus form to obtain a model

$$X = X_{\dot{u}}\dot{u} + X_u u + X_{|u|u}|u|u + X_{\dot{v}}\dot{v} + X_v v + X_{|v|v}|v|v + X_{\dot{w}}\dot{w} + X_w w + X_{|w|w}|w|w + \cdots .$$

More generally, any hydrodynamic load component is susceptible to be written as

$$X = \sum_i X_i \, \mathrm{g}_{X_i}(\nu_r, \dot{\nu}_r), \tag{2.14}$$

where the X_i are coefficients and g_{X_i} are basis functions of ν_r and $\dot{\nu}_r$. X_i coefficients will be further called *manoeuvring coefficients*. See *e.g.* [126, 98], where different options for model (2.14) have been explored.

2.2.4 A NOTE ON THE CABLE MODEL

Further on this work, cable modelling will be approached in greater detail. The main motivation of cable modelling is to compute a vector of the forces and moments exerted by the cable on the ROV, τ_{cable}, to be used on model (2.5). Vector τ_{cable} results from all dynamic phenomena on the cable, namely drag due to current, hydrostatics, spatial configuration and motion, among others. Figure 2.2 illustrates how, when the ROV is deployed, the cable obtains certain geometric configuration and, for instance, it accumulates the effect of current along its length. Finally, the sum of all effects is summarised in τ_{cable}.

2.2.5 ACTUATORS

The body's forces and moments vector due to thrusters, $\tau(\mathbf{u})$, is computed by knowing the location and orientation of each thruster, as well as the way each thruster force (*i.e.*, thrust) behaves; τ is dependent on the input vector $\mathbf{u}$. Supposing an ROV is composed by r thrusters, then $\mathbf{u} \in \mathbb{R}^r$ and $\mathbf{u} = \begin{bmatrix} u_1 & \cdots & u_i & \cdots & u_r \end{bmatrix}^{\mathsf{T}}$.

The location of each thruster is given by $\mathbf{r}^b_{t_i/b} \in \mathbb{R}^3$, $i = 1, \ldots, r$, and its orientation can be expressed by using a unitary vector $\mathbf{e}^b_{t_i/b} \in \mathbb{R}^3$. Each thruster force is given by $f_i \in \mathbb{R}$, and the body's forces and moments vector is given by $\tau(\mathbf{u}) = \sum_{i=1}^{r} \tau_i(u_i)$, where the contribution of each thruster is

$$\tau_i(u_i) = \begin{bmatrix} \mathbf{F}_{\tau_i}(u_i) \\ \mathbf{M}_{\tau_i}(u_i) \end{bmatrix} = \begin{bmatrix} \mathbf{e}^b_{t_i/b} f_i(u_i) \\ \mathrm{S}\left(\mathbf{r}^b_{t_i/b}\right) \mathbf{e}^b_{t_i/b} f_i(u_i) \end{bmatrix} = \begin{bmatrix} \mathbf{e}^b_{t_i/b} \\ \mathrm{S}\left(\mathbf{r}^b_{t_i/b}\right) \mathbf{e}^b_{t_i/b} \end{bmatrix} f_i(u_i).$$

Vectors $\mathbf{r}^b_{t_i/b}$, $\mathbf{e}^b_{t_i/b}$, and $\mathbf{e}^b_{t_i/b} f_i(u_i)$ are illustrated in Fig. 2.3 for PIONERO500. This ROV has a total of six thrusters, and Fig. 2.3 shows vectors for thruster 2 explicitly. Vector $\mathbf{r}^b_{t_2/b}$ indicates

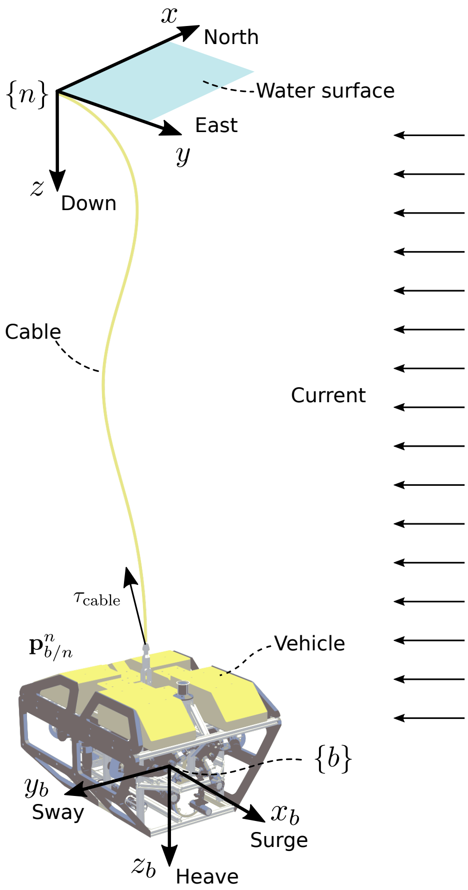


Figure 2.2. ROV and cable. This figure illustrates a situation where the ROV has been deployed and attains certain depth. If there is a current profile, this effect should accumulate along the cable's length and affect ROV motion. The figure was created by the author from PIONERO500's CAD model and the CAD render was obtained thanks to Sergio Suárez.

the position of any point in thruster 2's force line of action. This line of action's direction is given by unitary vector $\mathbf{e}^b_{t_{2/b}}$. Finally, the complete force vector becomes $\mathbf{e}^b_{t_{2/b}} f_2$. This whole process may be repeated for all remaining thrusters.

Each thruster body load can be written more compactly as

$$\tau_i(u_i) = \mathbf{T}_i f_i(u_i).$$

where

$$\mathbf{T}_i = \left[\begin{array}{c} \mathbf{e}^b_{t_{i/b}} \\ \mathrm{S}\left(\mathbf{r}^b_{t_{i/b}}\right) \mathbf{e}^b_{t_{i/b}} \end{array} \right]. \tag{2.15}$$

Moreover, the contribution of all thrusters may be computed by defining the matrix-vector

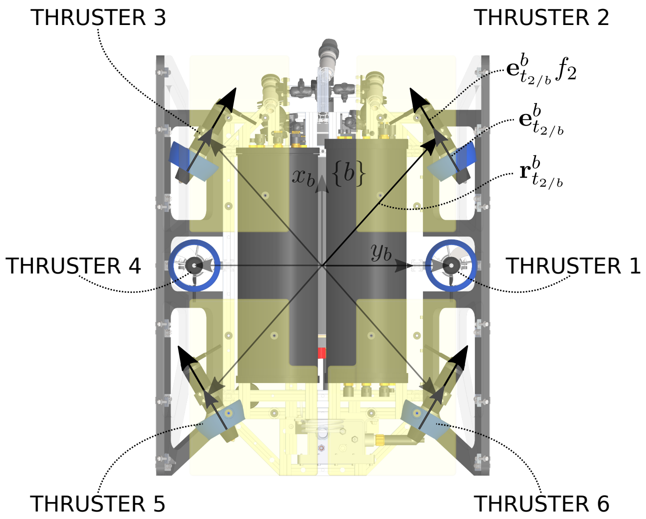


Figure 2.3. Pionero500's thruster configuration. The ROV has six thrusters but related vectors are illustrated for thruster 2 only; related vectors indicate thrust force and the unitary vector of its line of action, as well as a position vector of any point along the line of action. As before, the figure was created by the author from Pionero500's CAD model and the CAD render was obtained thanks to Sergio Suárez.

product [36]

$$\tau(\mathbf{u}) = \mathbf{T}\mathbf{f}(\mathbf{u}), \tag{2.16}$$

where $\mathbf{T} \in \mathbb{R}^{6\times r}$ is called thruster configuration matrix, $\mathbf{f} = \begin{bmatrix} f_1 & \cdots & f_i & \cdots & f_r \end{bmatrix}^\mathsf{T}$ is the vector of thruster forces. The vector of thruster forces defines a space in $\mathbb{R}^r$. The thruster configuration matrix is assembled as

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 & \cdots & \mathbf{T}_i & \cdots & \mathbf{T}_r \end{bmatrix}, \tag{2.17}$$

where each column is computed from (2.15). Consequently, (2.16) is a mapping from thruster forces space (in $\mathbb{R}^r$) to three-dimensional forces and moments space (in $\mathbb{R}^6$).

The force $f_i(u_i)$ exerted by each thruster, dropping sub-index i, is calculated as

$$f = \rho K_T(J) D^4 |n| n, \tag{2.18}$$

where D is propeller diameter, n propeller speed (often in s^{-1}), and $K_T(J)$ propeller's thrust coefficient as a function of the advance ratio J. The advance ratio is given by

$$J = \frac{V_a}{Dn}, \tag{2.19}$$

where V_a is advance velocity. In (2.18), each thruster's propeller speed n_i depends on input u_i, meaning that thrust is controlled by manipulating propeller speed. The relation $n_i(u_i)$

depends on the thruster's driver unit. Often, the input signal and propeller speed are proportional. In this case, $n_i = n_{i_{\max}} u_i$, where $n_{i_{\max}}$ is maximum propeller speed and u_i is a normalised input, *i.e.*, $u_i \in [-1, 1]$.

2.3 CONTROL SYSTEM

In this work, two main components of an ROV control system are considered, namely thrust allocation and control algorithms. These components are illustrated in Fig. 2.4 as generic blocks. Broadly speaking, the control algorithm requires converting some sort of operator commands into control commands. The control commands are often related to a body forces and moments vector in frame $\{b\}$ that controls motion in three-dimensional space, *i.e.*, about all six translation and rotation degrees of freedom. Then, the thrust allocation algorithm converts these control commands into commands to each thruster. Manual and feedback of control schemes are considered herein, including manual, multi-loop PID, and non-linear PID.

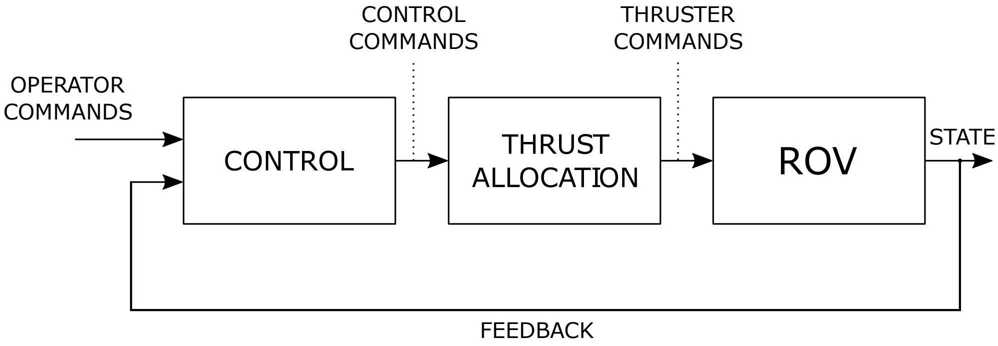


Figure 2.4. Control system. This block diagram shows that each control system converts operator commands into thrusters commands in order to attain certain motion. The control scheme could or could not have any means of feedback.

All feedback control algorithms depend on knowing the vehicle's state $\mathbf{q} = \begin{bmatrix} \eta^\intercal & \nu^\intercal \end{bmatrix}^\intercal$. This is often accomplished by using an algorithm that uses (noisy) measurements to estimate all state variables; often this estimator is based on the Kalman filter [48]. This work will assume that measurable information is available and a state estimator is constructable; consequently, this work will not deal with how to construct such estimator nor its effects on motion feasibility.

Additionally, the control system has the following simplifications and considerations:

- Further measurement-related dynamic effects are neglected, such as sensor and state estimator dynamics.
- All implementation issues are disregarded. It is assumed that control strategies are possible to implement in the ROV's control computer. Also, discretisation-related effects on controllers are not studied.
- High-level control strategies, such as mission and trajectory planning, are not considered.
- From all controllable degrees of freedom, yaw is the most important one to control, followed by heave [26].

2.3.1 THRUST ALLOCATION

Thrust allocation is the most basic action towards controlling an ROV, even when the vehicle is manually controlled and no feedback control strategy has been implemented. The task of thrust allocation is to translate motion commands in three-dimensional space into thruster commands [36]. This means finding a mapping from a vector

$$\tau_u = \begin{bmatrix} X_u & Y_u & Z_u & K_u & M_u & N_u \end{bmatrix}^\intercal,$$

representing motion commands in the forces and moments space, into a thrusters command vector $\mathbf{u} = \begin{bmatrix} u_1 & \cdots & u_i & \cdots & u_r \end{bmatrix}^\intercal$.

To find such map, Eq. (2.16) is often used to define an inverse map, *i.e.*, from three-dimensional forces and moments space to thruster forces space. Depending on the thrusters number and how they are distributed, the system may be under-actuated, fully actuated, or over-actuated. For matrix $\mathbf{T} \in \mathbb{R}^{6\times r}$ in (2.16), this can be determined from whether its rank is less, equal, or greater than the degrees of freedom. For an under-actuated system one can define a mapping that represents a fully actuated system on a reduced space, *e.g.*, $\mathbb{R}^r$. For an over-actuated system there is not a unique inverse map.

A common way to define an inverse map is by using the Moore-Penrose pseudo-inverse as

$$\mathbf{T}^\dagger = \mathbf{T}^\intercal(\mathbf{T}\mathbf{T}^\intercal)^{-1}.$$

With this pseudo-inverse, an inverse mapping is defined as

$$\mathbf{f}_u = \mathbf{T}^\dagger \tau_u. \tag{2.20}$$

This mapping computes the amount of thrust at all thrusters $\mathbf{f}_u$ from a τ_u that represents a command for the desired net body forces and moments vector, *i.e.*,

$$\tau_u = \begin{bmatrix} X_u & Y_u & Z_u & K_u & M_u & N_u \end{bmatrix}^\intercal.$$

Then, $\mathbf{f}_u$ is a vector of thrust commands given by $\mathbf{f}_u = \begin{bmatrix} f_{u_1} & \cdots & f_{u_i} & \cdots & f_{u_r} \end{bmatrix}^\mathsf{T}$.

Once $\mathbf{f}_u$ is computed, it is necessary to compute the input to each thruster, *i.e.*, all u_i in (2.16). To improve linearity of the control loop, it is convenient to acknowledge in (2.18) that thrust is proportional to the square of the propeller's velocity, as well as that propeller's velocity is often proportional to control input. Then, from (2.18)

$$n = \text{sign}(f)\sqrt{\left|\frac{1}{\rho K_T(J) D^4} f\right|}$$

and considering that $n_i = n_{i_{\max}} u_i$, one could define the expression for the thruster input u_i as

$$u_i = \frac{\text{sign}(f_{u_i})}{n_{\max_i}} \sqrt{\left|\frac{1}{\rho K_T D^4} f_{u_i}\right|}, \tag{2.21}$$

where choosing $K_T = K_{T_0} = K_T(0)$, the thrust coefficient at bollard pull, is customary. This map $\mathbf{f}_u \mapsto \mathbf{u}$ is represented by

$$\mathbf{u} = \mathbf{f}_u^\dagger(\mathbf{f}_u). \tag{2.22}$$

Finally, as one could notice, because τ_u are force/moments commands and are expressed in force/moments units, it is convenient to compute such values from normalised inputs. This can be written, for each force/moment, as $X_u = X_{\max} u_X$, $Y_u = Y_{\max} u_Y$, $Z_u = Z_{\max} u_Z$, $K_u = K_{\max} u_K$, $M_u = M_{\max} u_M$, and $N_u = N_{\max} u_N$. Please note that all $u_{(\cdot)} \in [-1, 1]$ and all terms like $X_{\max}$ are a constant that represents the maximum force/moment. This defines the normalised input command

$$\mathbf{u}_c = \begin{bmatrix} u_X & u_Y & u_Z & u_K & u_M & u_N \end{bmatrix}^\mathsf{T}.$$

Figure 2.5 illustrates the aforementioned thrust allocation algorithm and how all pieces come up together.

2.3.2 CONTROL

Three types of control are considered: manual, multiloop PID, and non-linear PID. In manual control all degrees of freedom are manipulated by the operator. This means that no feedback from measurements is used; the only required component is thrust allocation. In a multi-loop PID, some or all degrees-of-freedom use feedback control, either to maintain any degree-of-freedom at a constant value or to operate in a fly-by-wire fashion. In a non-linear PID, all degrees of freedom are feedback-controlled in order to execute previously established trajectories or keep position in the NED frame $\{n\}$.

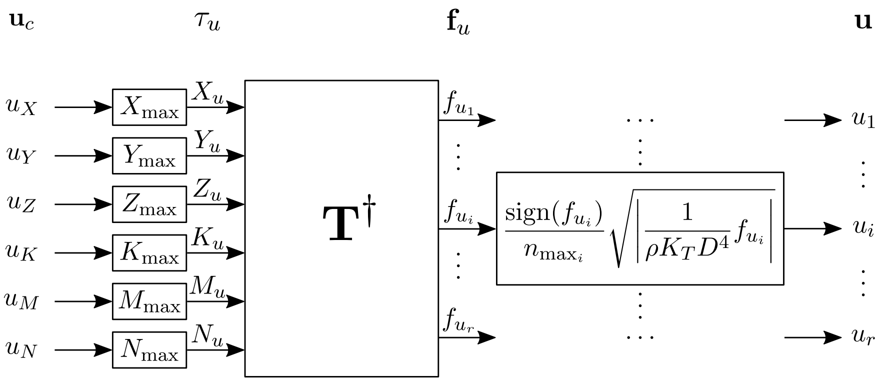


Figure 2.5. Simple thrust allocation algorithm

2.3.2.1 Open-loop control

Open-loop or manual control is an algorithm that converts operator commands into thruster commands without any feedback from (motion) sensors. The commonly controlled degrees of freedom on an ROV are surge, sway, heave, and yaw, *i.e.*, three-dimensional translation and heading. This means that manual control requires a joystick or equivalent to capture signals j_X, j_Y, j_Z, and j_N, corresponding to the aforementioned degrees-of-freedom. In this case, joystick signals j_X, j_Y, j_Z, and j_N are processed to compute u_X, u_Y, u_Z, and u_N respectively, *i.e.*, control commands. Signal processing often includes filtering and normalisation. This processing is represented by functions $F_X(j_X)$, $F_Y(j_Y)$, $F_Z(j_Z)$, and $F_N(j_N)$. Moreover, for degrees-of-freedom roll and pitch one assumes $u_K = u_M = 0$; these degrees-of-freedom are not controlled and naturally stable. These ideas are illustrated as block diagrams in Fig. 2.6.

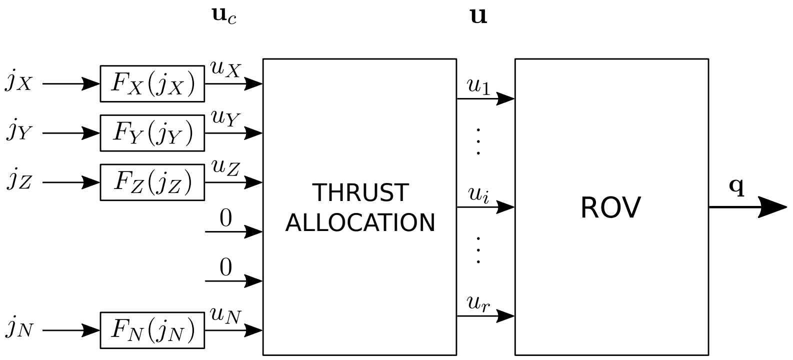


Figure 2.6. Manual control algorithm

2.3.2.2 Multi-loop PID

The multi-loop PID is a simple multi-variable feedback control strategy. It is a rather flexible control scheme because it uses a number of PID algorithms in an ad hoc configuration or structure. The basic PID algorithm used herein is

$$u(t) = \mathrm{PID}(y_{\mathrm{sp}}, y(t)) = K_P(y_{\mathrm{sp}} - y(t)) + K_I \int_0^t (y_{\mathrm{sp}} - y(\mathrm{t}))\mathrm{dt} - K_D \dot{y}(t),$$

where y is the controlled variable; y_{sp} set-point; u control command; and K_P, K_I, and K_D tuning parameters.

A straight-forward approach to control an ROV is to use a PID controller for each controlled degree of freedom. As mentioned before, the most commonly controlled degrees of freedom are those related to surge, sway, heave, and yaw. In this case, a multi-loop PID is defined to control surge velocity u, sway velocity v, depth/altitude z and heading ψ. A diagram of this multi-loop PID is shown in Fig. 2.7.

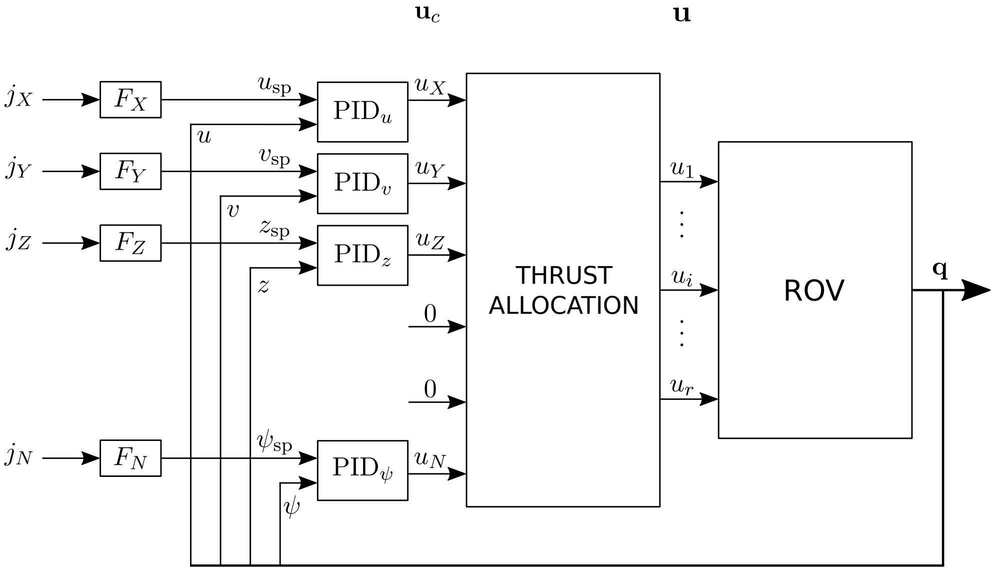


Figure 2.7. Simple multiloop PID

More specifically, surge and sway PIDs, identified as $\mathrm{PID}_u(u_{\mathrm{sp}}, u)$ and $\mathrm{PID}_v(v_{\mathrm{sp}}, v)$ respectively, are used to control surge and sway velocities u and v in order to operate in a fly-by-wire fashion. Joystick commands $j_X$and j_Y are then related to their respective velocity set-points. Heave PID, identified as $\mathrm{PID}_z(z_{\mathrm{sp}}, z)$, is used to control vertical position. Vertical position could be depth or altitude. In this PID, the derivative of the controlled variable, $\dot{z}$, could be replaced by heave velocity w, *i.e.*, $\dot{z} = w$. Joystick command j_Z is then related to the desired depth or altitude. Yaw PID, identified as $\text{PID}_\psi(\psi_{\text{sp}}, \psi)$, is used to control heading. In this PID, the derivative of the controlled variable, $\dot{\psi}$, could be replaced by yaw angular rate r, *i.e.*, $\dot{\psi} = r$. Joystick command j_N is then related to the desired heading.

Taking into account this multi-loop PID structure, a transition from manual control to feedback control could be made by adding PIDs one-by-one. Given that yaw is the most important degree-of-freedom to control, followed by heave [26], a heading-depth multi-loop PID controller is illustrated in Fig. 2.8. In this case, there is feedback control on yaw and heave and manual control on surge and sway velocities.

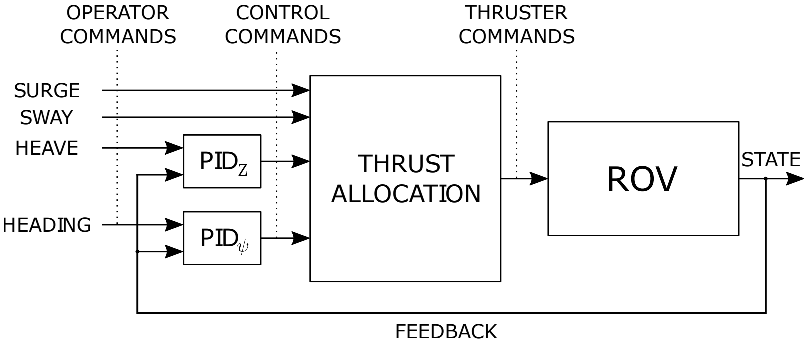


Figure 2.8. Heading-depth multiloop PID

2.3.2.3 Non-linear PID

Feedback control is often approached by using a non-linear PID scheme. In this case, all degrees of freedom are feedback-controlled in order to execute previously established trajectories or keep position in the NED frame $\{n\}$; this means controlling ROV's configuration (position and attitude) η and its derivative $\dot{\eta}$. But, because ROV's net forces and moments are defined in body frame $\{b\}$, defining the control law requires using kinematic transformation $J(\eta)$.

The non-linear PID control law is then

$$\mathbf{u}_c = -\mathbf{J}(\eta)^{-1}\left(\mathbf{K}_p\tilde{\eta} + \mathbf{K}_d\dot{\tilde{\eta}} + \mathbf{K}_i\int_0^t \tilde{\eta}(\mathrm{t})d\mathrm{t}\right), \tag{2.23}$$

where $\tilde{\eta} = \eta - \eta_d$ is the error and η_d a reference value related to operator commands. Here, a regular PID scheme is used to define an intermediate control command in NED space, then inverse transformation $J^{-1}(\eta)$ is used to map this intermediate control command into body frame $\{b\}$, to finally compute control command $\mathbf{u}_c$. A block diagram of the control scheme is illustrated in Fig. 2.9.

Put differently, Fig. 2.10 illustrates the idea of non-linear PID control in the ROV's motion

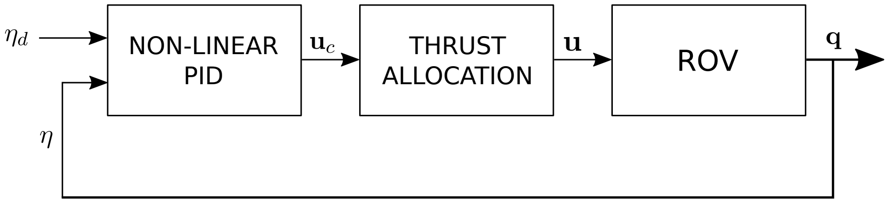


Figure 2.9. Non-linear PID block diagram

space. Here, configuration η is given by the position and orientation of frame $\{b\}$ with respect to frame $\{n\}$. The figure shows two configurations: actual configuration η and desired configuration η_d. The difference between those two configurations is the error, given by $\tilde{\eta} = \eta - \eta_d$. If a PID algorithm is computed from η and $\dot{\eta}$, control action $\mathbf{K}_p\tilde{\eta} + \mathbf{K}_d\dot{\tilde{\eta}} + \mathbf{K}_i \int_0^t \tilde{\eta}(\mathrm{t})dt$ is defined in $\{n\}$ space. By using inverse transformation $\mathbf{J}^{-1}(\eta)$ this action is transformed into $\{b\}$ space.

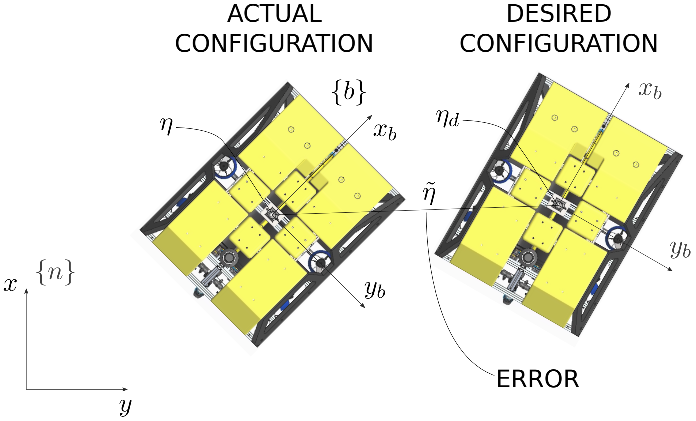


Figure 2.10. Illustration of non-linear PID in the ROV

Chapter 3

Study subjects

Throughout this thesis, three ROVs are used as study subjects, namely UPB's visor3, UPB's Pionero500, and NTNU's Minerva. The three ROVs span the observation and mid-size classes and offer different types of information. Because UPB's ROVs are in-house developments, they will serve as study subjects for developing models from scratch, as well as illustrating the different procedures and methodologies proposed in this thesis. Particularly, because Pionero500 is under development it will be useful to carry through design-related analyses. Conversely, because Minerva was industrially produced and has been widely tested at NTNU, available models will be useful for directly illustrating the different procedures and methodologies without further modelling. This chapter describes the aforementioned ROVs and presents information useful for performing computations on models such as (2.5).

The required modelling information focuses around the properties necessary to compute mass and inertia, hydrostatics, hydrodynamics, actuators, and thrust allocation. To model mass and inertia, one requires properties such as mass, moments of inertia, and coordinates of the centre of mass (or centre of gravity). To model hydrostatics, one requires information about weight and payload, as well as centre of buoyancy and any means to compute buoyancy force, like knowing the ROV's volume and water's density. In order to model hydrodynamics, one requires parameters for, *e.g.*, quadratic/linear drag and added mass. Nevertheless, because hydrodynamics will be approached later on, this subject is treated here rather superficially.

Furthermore, in order to model actuators, it is necessary to know each thruster's location and orientation. This may be divided into propeller and motor/drive information. The propeller information may be propeller type, diameter, and pitch-diameter ratio or any other information useful to compute or estimate thrust coefficient. The motor/drive information is often related to knowing the limitations it poses on the propeller, such as maximum rotational speed. Finally, because the possibilities are many, for each ROV one alternative to the thrust allocation algorithm will be presented.

3.1 VISOR3

VISOR3, depicted in Fig. 3.1, is an experimental, observation-class ROV developed in UPB during the late 2000's. This ROV was first developed for surveillance and inspection of port facilities [100, 57], but, afterwards, it has been used as a test-bed for studying numerous ROV-related subjects [131, 98, 147, 8, 7, 106, 56, 92, 31, 133]. It has capability for capturing video and measuring pressure (depth) and temperature [57].

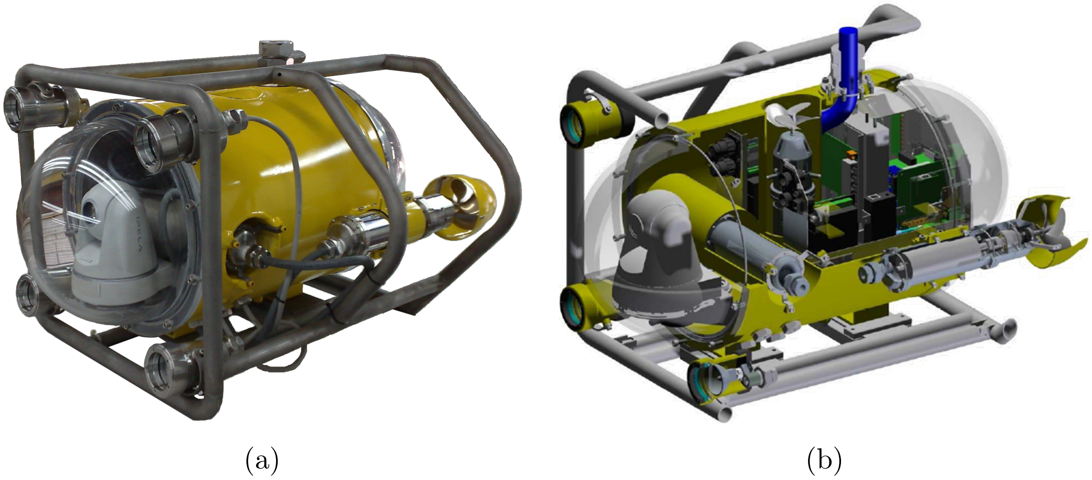

(a) (b)

Figure 3.1. VISOR3 ROV: (a) picture and (b) CAD. These images correspond to ca. 2009 version of this ROV. Since then, VISOR3 has been subject to many updates.

VISOR3's mass and volume properties are estimated from experimental and computed information. Moments of inertia (radii of gyration), position of the centre of mass, and position of the centre of buoyancy, as well as a first estimation of the volume and mass, are calculated using a comprehensive CAD model. Then, mass and volume are corroborated through experiments. The geometry, and consequently mass and volume properties, has varied during the years. In this study, a fixed geometry and set of parameters are used. A summary of these parameters is shown in Table 3.1.

Table 3.1. VISOR3 mass and volume properties

Property	Variable	Value	Units
Mass	m	62.94	kg
Radius of gyration x	k_{xx}	0.2120	m
Radius of gyration y	k_{yy}	0.1969	m
Radius of gyration z	k_{zz}	0.2157	m
Vertical distance between c.o.m. and c.o.b.	r_g	0.02	m

As part of this thesis' preliminary works, a manoeuvring model was obtained using VISOR3's geometry [98]. The manoeuvring model is a meta-model assembled from CFD computations, using MARIN's RANS equations solver ReFRESCO. The manoeuvring model in component form has the structure

$$X = (X_{u|u|}|u| + X_{u|v|}|v| + X_{u|w|}|w|)u + X_{vr}vr + X_{u|r|}u|r|, \tag{3.1a}$$

$$Y = (Y_{|u|v}|u| + Y_{v|v|}|v| + Y_{v|w|}|w|)v + Y_{u|r|}u|r| + Y_{v|r|}v|r|, \tag{3.1b}$$

$$Z = (Z_{|u|w}|u| + Z_{|v|w}|v| + Z_{w|w|}|w|)w, \tag{3.1c}$$

$$K = K_p p + K_{p|p|}p|p| + (K_{vw}w + K_{v|w|}|w|)v, \tag{3.1d}$$

$$\begin{aligned} M = {} & M_q q + M_{q|q|}q|q| + (M_{|u|w}|u| + M_{w|w|}|w|)w + M_{v|w|}v|w| \\ & + M_{ab_1}(u^2 - w^2)uw|\sin(\alpha/2)| + M_{ab_2}|uw^2|\mathrm{sign}(uw), \end{aligned} \tag{3.1e}$$

$$\begin{aligned} N = {} & N_r r + N_{r|r|}r|r| + (N_{|u|v}|u| + N_{v|v|}|v| + N_{v|w|}|w|)v \\ & + N_{ab_1}(u^2 - v^2)uv|\sin(\beta/2)| + N_{ab_2}|uv^2|\mathrm{sign}(uv) \\ & + N_{vr}vr + N_{|ur|}|ur|\mathrm{sign}(r). \end{aligned} \tag{3.1f}$$

In this model, terms such as $X_{u|u|}$, $X_{u|v|}$, $X_{u|w|}$, and so on are the manoeuvring coefficients. This structure is chosen so that the model captures the variation obtained from CFD computations; this is shown in [98]. The coefficients are specified in Table 3.2.

VISOR3's propulsion system has four thrusters to control four degrees of freedom, namely surge, sway, heave, and yaw. The general distribution is illustrated in Fig. 3.2. This configuration gives all three degrees of freedom in the horizontal plane; conversely, the fourth degree of freedom is heave (vertical motion). All thrusters have fixed position and orientation, and motion control is obtained by varying the propeller's velocity as follows:

- Two longitudinal thrusters, located port and starboard, are combined to control both surge and yaw.
- One lateral tunnel thruster controls sway. It is slightly displaced towards the bow with respect to the geometrical centre of the ROV. This and the two previous thrusters are able to fully control motion in the horizontal plane.

Table 3.2. Manoeuvring coefficients

Coeff.	Value	Coeff.	Value	Units
$X_{u\|u\|}$	-72.5609	$Y_{v\|v\|}$	-78.1767	N/(m/s)^2
$Z_{w\|w\|}$	-95.8881	$X_{u\|v\|}$	-21.5006	
$X_{u\|w\|}$	-34.0911	$Y_{\|u\|v}$	-67.0239	
$Y_{v\|w\|}$	-24.9222	$Z_{\|u\|w}$	-77.8589	
$Z_{\|v\|w}$	-47.4107			
K_{vw}	-3.8601	$M_{u\|w\|}$	-1.8696	Nm/(m/s)^2
$M_{w\|w\|}$	-2.6395	$M_{v\|w\|}$	-0.5831	
$N_{\|u\|v}$	-0.3147	$N_{v\|v\|}$	-3.4828	
$N_{v\|w\|}$	-2.0364			
$X_{u\|r\|}$	-78.1726	X_{vr}	-3.8179	N/(m rad/s^2)
$Y_{v\|r\|}$	-89.4249	$Y_{u\|r\|}$	-11.5705	
N_{vr}	-3.9045	$N_{\|ur\|}$	-1.1103	N/(rad/s^2)
M_{ab_1}	5.7699	N_{ab_1}	-4.8905	Nm/(m/s)^4
M_{ab_2}	-10.3963	N_{ab_2}	-5.9839	Nm/(m/s)^3
K_p	-0.3894	M_q	-0.9437	Nm/(rad/s)
N_r	-1.7661			
$K_{p\|p\|}$	-1.5274	$M_{q\|q\|}$	-1.5478	Nm/(rad^2/s^2)
$N_{r\|r\|}$	-1.0540			

- One vertical tunnel thruster controls heave.
- Roll and pitch are not controlled: the hydrostatic restoring moments allow the ROV to be naturally stable in these two degrees of freedom.

The parametrisation used to compute thruster position and orientation is shown in Table 3.3, where $r_{H_x} = 0.22$ m, $r_{H_y} = 0.25$ m, $r_{H_z} = 0.185$ m, $r_{T_x} = 0.414$ m, and $r_{T_z} = 0.185$ m. Table 3.3 shows also the parameters required to estimate thrust coefficient.

VISOR3's thrusters were developed in-house. This means that the limitations imposed by the motors, such as maximum propeller speed, are not known beforehand and must be computed. Each thruster has a brushless DC motor and gearbox, as well as required electronics and drivers. All thrusters use the same motor-reducer combination. To compute the limitations imposed by the motor-propeller combination, a model for electrical/mechanical power conversion in steady-state is considered. This model considers that available power is used to produce motion and counteract propeller's torque. Also, it is considered that energy is not

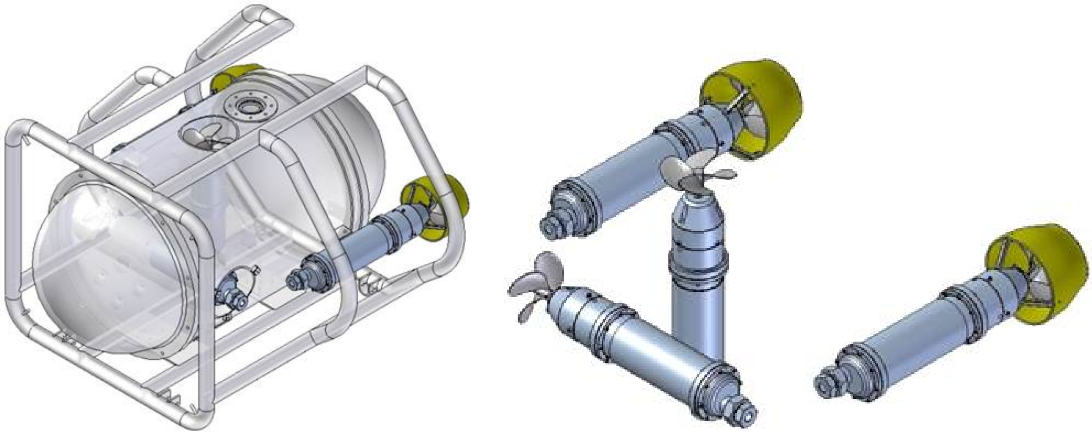


Figure 3.2. Thrusters distribution

Table 3.3. Propeller attributes

Parameter	Starboard	Port	Transversal	Vertical
Propeller diameter (mm)	101.6	101.6	91.44	91.44
Number of blades	4	4	4	4
Pitch-diameter ratio	1.275	1.275	1.0	1.0
Position x (m)	$-r_{H_x}$	$-r_{H_x}$	r_{T_x}	0
Position y (m)	r_{H_y}	$-r_{H_y}$	0	0
Position z (m)	r_{H_z}	r_{H_z}	r_{T_z}	0
Unitary vector	$[\ 1 \quad 0 \quad 0\ ]^\intercal$	$[\ 1 \quad 0 \quad 0\ ]^\intercal$	$[\ 0 \quad -1 \quad 0\ ]^\intercal$	$[\ 0 \quad 0 \quad 1\ ]^\intercal$

accumulated neither by inertia nor inductance. The model could be written as

$$\omega = \frac{1}{Z}\left(n_0 u - \frac{1}{Z\eta}\frac{\mathrm{d}n}{\mathrm{d}M}\mathrm{Q}(\omega, V_a)\right),$$

where ω is propeller's angular speed, V_a advance velocity, n_0 motor's no-load speed, u driver's normalised control signal ($\in [-1, 1]$), $\mathrm{d}n/\mathrm{d}M$ motor's speed/torque gradient (assumed constant), $\mathrm{Q}(\omega, V_a)$ propeller's torque, Z reducer's ratio, and η reducer's efficiency. The propeller's torque is computed as $Q = \rho K_Q(J) D^5 |\omega| \omega$, where $K_Q(J)$ is torque coefficient and $J = V_a/(D\omega)$ is advance ratio. All motor-reducers use the parameters shown in Table 3.4.

Thrust allocation for VISOR3 fits a four-degree-of-freedom formulation. Thus, the vector of thrust forces is $\mathbf{f} = [\ f_1 \quad f_2 \quad f_3 \quad f_4\ ]^\intercal$, representing starboard, port, transversal, and vertical thrusters respectively. This four-degree-of-freedom formulation eliminates non-controlled, naturally stable pitch and roll motions. For this ROV's thrusters position and orientation,

Table 3.4. Motor-reducer parameters

Parameter	Value	Units
n_0	5952	rpm
dn/dM	6.16	rpm/mNm
Z	3.5	-
η	0.8	-

map (2.17) is given by

$$\begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_K \\ \tau_M \\ \tau_N \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & r_{T_z} & 0 \\ r_{H_z} & r_{H_z} & 0 & 0 \\ r_{H_y} & -r_{H_y} & -r_{T_x} & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

From this complete thruster configuration mapping a dimension-reduced mapping may be obtained. In this reduced mapping, roll and pitch degrees-of-freedom are eliminated. Thus,

$$\begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_N \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ r_{H_y} & -r_{H_y} & -r_{T_x} & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

An inverse mapping computed from the dimension-reduced mapping becomes

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} 1/2 & -r_{T_x}/(2r_{H_y}) & 0 & 1/(2r_{H_y}) \\ 1/2 & r_{T_x}/(2r_{H_y}) & 0 & -1/(2r_{H_y}) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_N \end{bmatrix}$$

This mapping represents Eq. (2.20), one part of the complete thrust allocation algorithm.

3.2 PIONERO500

Pionero500 is UPB's latest ROV development and the first one in the mid-size range. It is rated to operate up to 500-m depth. Currently, it is being developed for offshore exploration, and, as of mid-2018, it had begun the test stages. It is equipped with three cameras, lighting system, CTD, and navigation sensors such as IMU, altimeter, and USBL.

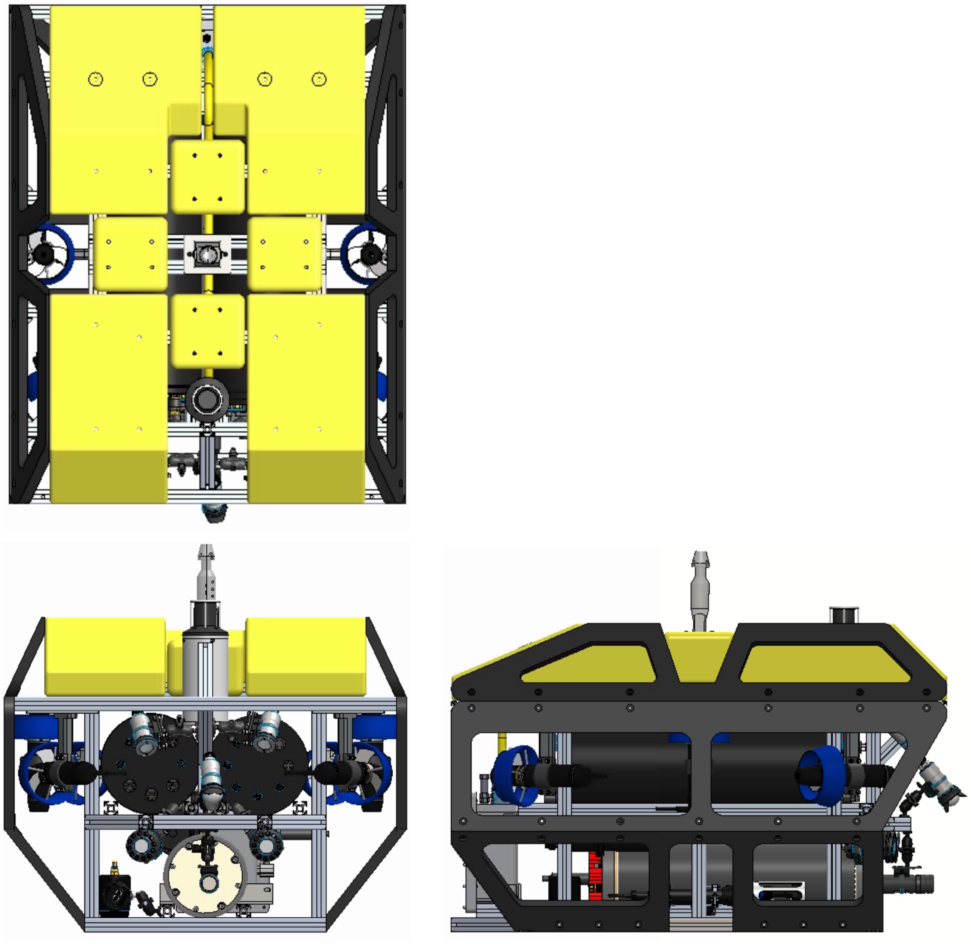


Figure 3.3. Pionero500 CAD

Pionero500's overall dimensions are $l = 1.26$ m, $b = 1.00$ m, and $h = 0.78$ m. The mass comes from measurements and is 264 kg. The moments of inertia and centre of mass are computed by using a comprehensive CAD model. A summary of these parameters is shown in Table 3.5.

A hydrodynamic model for Pionero500 is not available beforehand. Nevertheless, an initial estimate of drag may be computed from assuming the drag coefficient of a rectangle, *i.e.*, $C_D = 1.2$. From this assumption an uncoupled drag matrix is defined as

$$\mathbf{D} = \frac{1}{2}\rho_{\mathrm{W}} C_D \,\mathrm{diag}\{bh|u|,\, lh|v|,\, lb|w|,\, 0,\, 0,\, 0\}.$$

Please note that velocities in this drag matrix must be relative and the relative sub-index is dropped.

Pionero500's propulsion system has six thrusters to control four degrees of freedom: surge,

Table 3.5. PIONERO500 mass and volume properties

Property	Variable	Value	Units
Mass	m	264	kg
Radius of gyration x	k_{xx}	0.1767	m
Radius of gyration y	k_{yy}	0.2379	m
Radius of gyration z	k_{zz}	0.1584	m
Radius of gyration x-y	k_{xy}	0.0031	m
Radius of gyration x-z	k_{xz}	-0.0150	m
Radius of gyration y-z	k_{yz}	0.0010	m
Position of the centre of mass	$\mathbf{r}^b_{g/b}$	$\begin{bmatrix} 0 & 0 & -0.23 \end{bmatrix}^\intercal$	m

sway, heave, and yaw. The general distribution is illustrated in Fig. 3.4; please note how thrusters are numbered. This configuration gives all three degrees of freedom in the horizontal plane; the fourth degree of freedom is heave (vertical motion). All these thrusters have fixed position and orientation. Motion control is obtained by varying the propeller's velocity as follows:

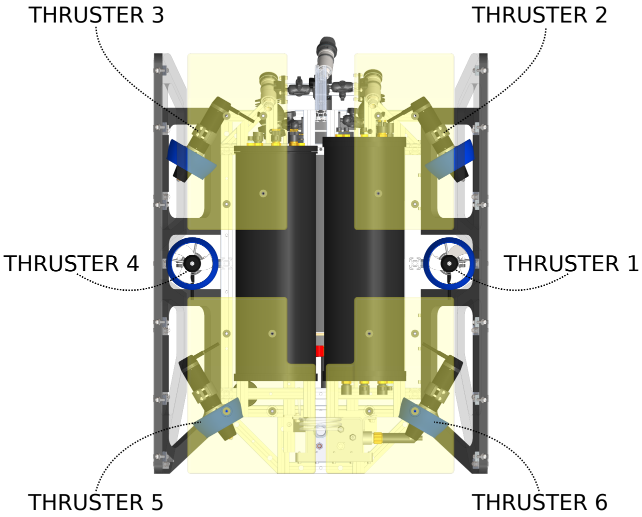


Figure 3.4. PIONERO500's thruster distribution. The figure was created by the author from PIONERO500's CAD model and the CAD render was obtained thanks to Sergio Suárez.

- Four thrusters, numbered 2, 3, 5, and 6 in Fig. 3.4, are located in the horizontal plane to jointly control surge, sway, and yaw.
- Two vertical thrusters, numbered 1 and 4 in Fig. 3.4, control heave. These two thrusters are assumed to operate redundantly.

- Roll and pitch are not controlled. As usual, the hydrostatic restoring moments allow the ROV to be naturally stable in these two degrees of freedom.

All thrusters used in PIONERO500 are the same model. The thrusters parameters are shown in Fig. 3.6. The parametrisation used to compute thruster position and orientation is shown in Table 3.7, where $r_{H_x} = 0.387$ m, $r_{H_y} = 0.348$ m, $r_{V_y} = 0.414$ m, and $\alpha = 30$ deg.

Table 3.6. Technadyne thrusters attributes

Parameter	Parameter
Propeller diameter (mm)	120.7
Forward bollard pull (N)	146.8
Backward bollard pull (N)	85.85
Maximum speed (rpm)	3000

Table 3.7. PIONERO500 thrusters distribution

Parameter	T 1	T 2	T 3	T 4	T 5	T 6
Position x	0	r_{H_x}	r_{H_x}	0	$-r_{H_x}$	$-r_{H_x}$
Position y	r_{V_y}	r_{H_y}	$-r_{H_y}$	$-r_{V_y}$	$-r_{H_y}$	r_{H_y}
Position z	0	0	0	0	0	0
Unitary vector x	0	$\cos\alpha$	$\cos\alpha$	0	$\cos\alpha$	$\cos\alpha$
Unitary vector y	0	$-\sin\alpha$	$\sin\alpha$	0	$-\sin\alpha$	$\sin\alpha$
Unitary vector z	1	0	0	1	0	0

Thrust allocation for PIONERO500 fits a formulation where four-degree-of-freedom motion is obtained through controlling six actuators. Thus, $\mathbf{f} = [\ f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad f_6\ ]^\intercal$ is the vector of thrust forces that represents thrusters as numbered in Fig. 3.4. For this ROV, map (2.17) is given by

$$\begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_K \\ \tau_M \\ \tau_N \end{bmatrix} = \begin{bmatrix} 0 & \cos\alpha & \cos\alpha & 0 & \cos\alpha & \cos\alpha \\ 0 & -\sin\alpha & \sin\alpha & 0 & -\sin\alpha & \sin\alpha \\ 1 & 0 & 0 & 1 & 0 & 0 \\ r_{V_y} & 0 & 0 & r_{V_y} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -r_{zz} & r_{zz} & 0 & r_{zz} & -r_{zz} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}$$

where $r_{zz} = r_{H_x} \sin\alpha + r_{H_y} \cos\alpha$, representing z-moment equivalent radius. From this, if one assumes that forces due to vertical thrusters satisfy $f_1 = f_4 = f_{14}$, the mapping may be rewritten as

$$\begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_K \\ \tau_M \\ \tau_N \end{bmatrix} = \begin{bmatrix} 0 & \cos\alpha & \cos\alpha & \cos\alpha & \cos\alpha \\ 0 & -\sin\alpha & \sin\alpha & -\sin\alpha & \sin\alpha \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -r_{zz} & r_{zz} & r_{zz} & -r_{zz} \end{bmatrix} \begin{bmatrix} f_{14} \\ f_2 \\ f_3 \\ f_5 \\ f_6 \end{bmatrix},$$

where, as one could notice, roll and pitch moments are eliminated. Consequently, a reduced mapping could be written as

$$\begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_N \end{bmatrix} = \begin{bmatrix} 0 & \cos\alpha & \cos\alpha & \cos\alpha & \cos\alpha \\ 0 & -\sin\alpha & \sin\alpha & -\sin\alpha & \sin\alpha \\ 2 & 0 & 0 & 0 & 0 \\ 0 & -r_{zz} & r_{zz} & r_{zz} & -r_{zz} \end{bmatrix} \begin{bmatrix} f_{14} \\ f_2 \\ f_3 \\ f_5 \\ f_6 \end{bmatrix}.$$

Then, the inverse mapping to define a thrust allocation algorithm, obtained from the thrust configuration matrix pseudo-inverse, is

$$\begin{bmatrix} f_{u_{14}} \\ f_{u_2} \\ f_{u_3} \\ f_{u_5} \\ f_{u_6} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1/2 & 0 \\ 1/(4\cos\alpha) & -1/(4\sin\alpha) & 0 & -1/(4r_{zz}) \\ 1/(4\cos\alpha) & 1/(4\sin\alpha) & 0 & 1/(4r_{zz}) \\ 1/(4\cos\alpha) & -1/(4\sin\alpha) & 0 & 1/(4r_{zz}) \\ 1/(4\cos\alpha) & 1/(4\sin\alpha) & 0 & -1/(4r_{zz}) \end{bmatrix} \begin{bmatrix} X_u \\ Y_u \\ Z_u \\ N_u \end{bmatrix}$$

or equivalently, by using the assumption $f_{u_1} = f_{u_4} = f_{u_{14}}$,

$$\begin{bmatrix} f_{u_1} \\ f_{u_2} \\ f_{u_3} \\ f_{u_4} \\ f_{u_5} \\ f_{u_6} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1/2 & 0 \\ 1/(4\cos\alpha) & -1/(4\sin\alpha) & 0 & -1/(4r_{zz}) \\ 1/(4\cos\alpha) & 1/(4\sin\alpha) & 0 & 1/(4r_{zz}) \\ 0 & 0 & 1/2 & 0 \\ 1/(4\cos\alpha) & -1/(4\sin\alpha) & 0 & 1/(4r_{zz}) \\ 1/(4\cos\alpha) & 1/(4\sin\alpha) & 0 & -1/(4r_{zz}) \end{bmatrix} \begin{bmatrix} X_u \\ Y_u \\ Z_u \\ N_u \end{bmatrix}.$$

This mapping represents Eq. (2.20) of the complete thrust allocation algorithm.

An alternative to this thrust allocation algorithm is to define a normalised version. For the normalised version the maximum body forces and moments are defined as (please refer to, *e.g.*, Fig. 2.5)

$$\begin{aligned} X_{\text{max}} &= \beta_X \left(4\cos\alpha\, K_T D^4 n_{\text{max}}^2\right), \\ Y_{\text{max}} &= \beta_Y \left(4\sin\alpha\, K_T D^4 n_{\text{max}}^2\right), \\ Z_{\text{max}} &= \beta_Z \left(2\, K_T D^4 n_{\text{max}}^2\right), \quad \text{and} \\ N_{\text{max}} &= \beta_N \left(4 r_{zz}\, K_T D^4 n_{\text{max}}^2\right). \end{aligned}$$

This definition assumes that maximum body load due to thrusters is a factor $\beta_{(\cdot)} \in [0,1]$ of the best case scenario of all possible thrusters contributing to each degree-of-freedom, at bollard pull and maximum speed. This given, the complete thrust algorithm may be written as

$$\begin{bmatrix} \hat{f}_{u_1} \\ \hat{f}_{u_2} \\ \hat{f}_{u_3} \\ \hat{f}_{u_4} \\ \hat{f}_{u_5} \\ \hat{f}_{u_6} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \beta_X & 0 & 0 & 0 \\ 0 & \beta_Y & 0 & 0 \\ 0 & 0 & \beta_Z & 0 \\ 0 & 0 & 0 & \beta_N \end{bmatrix} \begin{bmatrix} u_X \\ u_Y \\ u_Z \\ u_N \end{bmatrix}$$

and

$$u_i = \operatorname{sign}(\hat{f}_{u_i})\sqrt{|\hat{f}_{u_i}|},$$

for $i = 1, \ldots, 6$. Factors $\beta_{(\cdot)}$ weigh the amount of power each degree-of-freedom receives from thrusters. This normalised algorithm is more convenient for implementation.

3.3 MINERVA

Minerva is NTNU's Sperre SUB-fighter 7500 ROV (see Fig. 3.5), used for biological and archaeological research and sampling, as well as for research on marine cybernetics. It is rated at a 700 m working depth, has a 600 m tether for power and communications, and is usually deployed from NTNU's research vessel (R/V) Gunnerus. For a more complete description, see for instance [122]. The model used within this thesis has the same parameters that were used by NTNU's AUR-Lab Minerva simulator during boreal fall 2016.

Minerva's overall dimensions are $l = 1.44$ m, $b = 0.82$ m, and $h = 0.80$ m, and its mass is 460 kg. The moments of inertia are computed by assuming the radii of gyration of a box, this is $k_{xx} = \frac{1}{12}\left(b^2 + h^2\right)$, $k_{yy} = \frac{1}{12}\left(l^2 + h^2\right)$, and $k_{zz} = \frac{1}{12}\left(b^2 + l^2\right)$. The properties are summarised in Fig. 3.8.

Hydrodynamics model assumes constant, diagonal added mass, and linear and quadratic drag. The added mass matrix is

$$\mathbf{M}_A = -\operatorname{diag}\{293., 302., 330., 110., 56., 55.\},$$

linear drag matrix is

$$\mathbf{D}_L = -\operatorname{diag}\{29., 41., 254., 34., 59., 45.\},$$

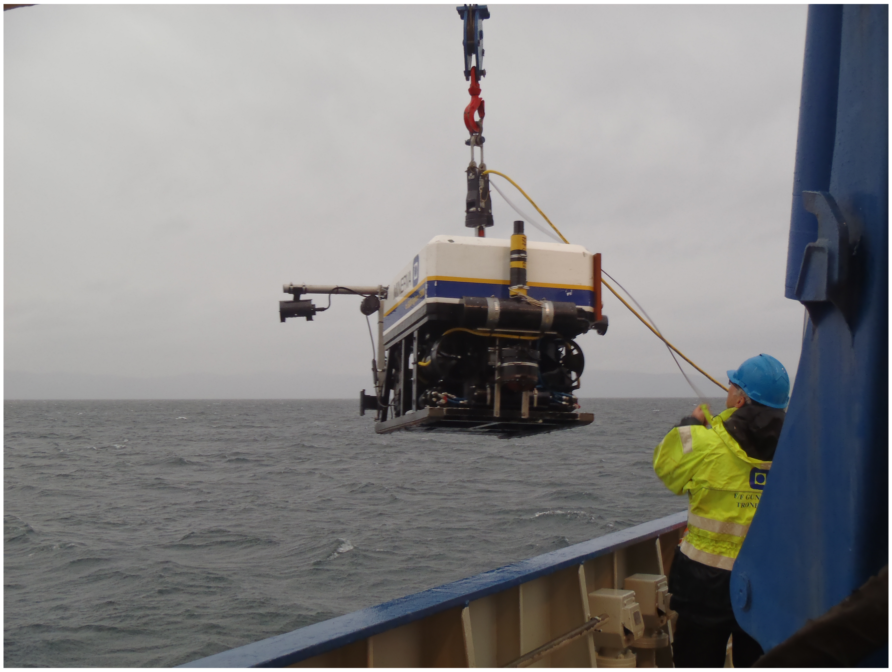


Figure 3.5. NTNU's Minerva, during an operation in Trondheim's Fjord during fall 2014. The picture was taken by Elkin Taborda and is being used with his permission.

and quadratic drag matrix is

$$\mathbf{D}_{NL} = -\,\mathrm{diag}\{292.|u|, 584.|v|, 635.|w|, 84.|p|, 148.|q|, 100.|r|\}.$$

Minerva's propulsion system has five thrusters to control four degrees of freedom, namely surge, sway, heave, and yaw. Its thrusters configuration gives all three degrees of freedom in the horizontal plane; the fourth degree of freedom is heave (vertical motion). Motion control is obtained by varying the propeller's velocity as follows:

- Two unidirectional, longitudinal thrusters, located port and starboard, are combined to control both surge and yaw. These thrusters are slightly rotated by an angle $\alpha = 10$ deg.
- One bidirectional, lateral thruster controls sway. It is slightly displaced towards the bow with respect to the geometrical centre of the ROV. This and the two previous thrusters are able to fully control motion in the horizontal plane.
- Two unidirectional, vertical thrusters control heave. These thrusters are assumed to operate redundantly.
- Roll and pitch are not controlled. As in the other ROVs, hydrostatic restoring moments allow the ROV to be naturally stable in these two degrees of freedom.

The parameters used to compute thruster position and orientation and propeller thrust coef-

Table 3.8. MINERVA mass and volume properties

Property	Variable	Value	Units
Mass	m	460	kg
Radius of gyration x	k_{xx}	0.1093	m
Radius of gyration y	k_{yy}	0.2261	m
Radius of gyration z	k_{zz}	0.2288	m
Position of the centre of gravity	r_g	0.12	m

ficient are shown in Table 3.9. The thrust coefficient for unidirectional thrusters is computed using a 1-3-quadrant formulation. The thrust coefficient in the first quadrant, where propeller and advance velocity are positive, is

Table 3.9. MINERVA's propeller attributes

Parameter	Starboard	Port	Lateral	Vertical-1	Vertical-2
Propeller diameter (mm)	220	220	190	220	220
Maximum speed (rpm)	1450	1450	1450	1450	1450
Position x (m)	-0.57	-0.57	0.166	0	0
Position y (m)	0.24	-0.24	0	0.2	-0.2
Position z (m)	0	0	0.3	0	0
Unitary vector x	$\cos(\alpha)$	$\cos(\alpha)$	0	0	0
Unitary vector y	$\sin(\alpha)$	$-\sin(\alpha)$	-1	0	0
Unitary vector z	0	0	0	1	1

$$K_T(J)_1 = 0.24J^3 - 0.25J^2 - 0.66J + 0.5.$$

The thrust coefficient in the third quadrant, where propeller and advance velocity are negative, is

$$K_T(J)_3 = 0.025J^3 - 0.28J^2 - 0.17J - 0.15.$$

For bidirectional thrusters first quadrant thrust coefficient is given by $K_T(J)_1$ and third quadrant thrust coefficient by $-K_T(J)_1$. Second and fourth quadrants, where propeller and advance velocities are different, are not considered because they are not easily predictable [36].

Thrust allocation for this ROV fits a formulation where four-degree-of-freedom motion is obtained through controlling five actuators. Thus, $\mathbf{f} = [\ f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5\ ]^\intercal$ is the vector of thrust forces that represents starboard, port, transversal, vertical-starboard, and vertical-port thrusters respectively. This formulation does not take into account non-controlled, naturally stable pitch and roll motions.

For this ROV, map (2.17) is given by

$$\begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_K \\ \tau_M \\ \tau_N \end{bmatrix} = \begin{bmatrix} \cos\alpha & \cos\alpha & 0 & 0 & 0 \\ \sin\alpha & -\sin\alpha & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ -r_{H_z}\sin\alpha & r_{H_z}\sin\alpha & -r_{L_z} & r_{V_y} & -r_{V_y} \\ r_{H_z}\cos\alpha & r_{H_z}\cos\alpha & 0 & 0 & 0 \\ -r_{zz} & r_{zz} & r_{L_x} & 0 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix},$$

where $r_{zz} = r_{H_x}\sin\alpha + r_{H_y}\cos\alpha$, representing z-moment equivalent radius due to starboard and port thrusters.

From this, if one assumes that forces due to vertical thrusters satisfy $f_1 = f_4 = f_{14}$ and, furthermore, roll and pitch degrees-of-freedom are eliminated, the mapping may be rewritten as

$$\begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_N \end{bmatrix} = \begin{bmatrix} \cos\alpha & \cos\alpha & 0 & 0 \\ \sin\alpha & -\sin\alpha & 1 & 0 \\ 0 & 0 & 0 & 2 \\ -r_{zz} & r_{zz} & r_{L_x} & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$$

Then, the inverse mapping is

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2\cos\alpha} & \frac{r_{L_x}}{2(r_{L_x}\sin\alpha + r_{zz})} & 0 & -\frac{1}{2(r_{L_x}\sin\alpha + r_{zz})} \\ \frac{1}{2\cos\alpha} & -\frac{r_{L_x}}{2(r_{L_x}\sin\alpha + r_{zz})} & 0 & \frac{1}{2(r_{L_x}\sin\alpha + r_{zz})} \\ 0 & \frac{r_{zz}}{r_{L_x}\sin\alpha + r_{zz}} & 0 & \frac{\sin\alpha}{2(r_{L_x}\sin\alpha + r_{zz})} \\ 0 & 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_N \end{bmatrix}$$

This mapping represents Eq. (2.20) of the complete thrust allocation algorithm.

Chapter 4

Tools from sphere geometry

In the chapters to come, the treatment of many concepts will be geometrical. This is stated for two reasons: first, model variables exist in some geometrical space and through their variations that space is explored; second, models are made up of mappings between two spaces, *i.e.*, to transform one geometrical space into another. Even though all variables treated here lie in Euclidean space, many variables are better defined by non-Euclidean subspaces. An example, of remarkable importance to this work, is the space of three-dimensional unitary vectors, which resembles a spherical surface. The problem of exploring this type of spaces is not trivial, if being efficient at that exploration is a concern (as it is when performing demanding computations). Altogether, this chapter's purpose is two-fold: introduce tools to deal with the problem of distributing of points in a spherical space and make a first application by performing graphical representations of some of this work's models.

4.1 SPHERICAL GEOMETRY

One of many examples where geometry concepts are useful is manoeuvring model (2.13): a mapping $\tau_H : \mathbb{R}^6 \times \mathbb{R}^6 \to \mathbb{R}^{12}$, that maps relative velocity and acceleration into a vector of forces and moments, *i.e.*, $(\nu_r, \dot{\nu}_r) \mapsto \tau_H$. (Please note that an abuse of notation is used: the same symbol is used to represent both the mapping and elements of the image set.) The spaces of ν_r and $\dot{\nu}_r$ are both six-dimensional. Focusing on ν_r and using the same concept of Eq. (2.1), a velocity vector is composed by both a linear and angular term. Assuming that these vectors represent velocities in body frame $\{b\}$, this derives in vectors $\mathbf{v}_r = \begin{bmatrix} u_r & v_r & w_r \end{bmatrix}^\mathsf{T} \in \mathbb{R}^3$ and $\omega = \begin{bmatrix} p & q & r \end{bmatrix}^\mathsf{T} \in \mathbb{R}^3$; hence, $\nu_r = \begin{bmatrix} \mathbf{v}_r & \omega \end{bmatrix}^\mathsf{T}$. This same idea could be applied to $\dot{\nu}_r$.

A convenient way to understand the behaviour of velocity components $\mathbf{v}_r$ and ω is by writing each vector as a multiplication of the magnitude and unitary vector. Further focusing on $\mathbf{v}_r$, one has $\nu_r = V\mathbf{e}_\nu$, where V is magnitude and $\mathbf{e}_\nu$ is a unitary vector. Geometrically, the space of unitary vectors such as $\mathbf{e}_\nu$ is a unitary sphere. Because the direction of relative velocities is important to this work, its geometry will be as well.

4.1.1 TWO-DIMENSIONAL CASE

In general, problems related to ROV motion are three-dimensional. Nevertheless, surface vessels or ROV planar motion, for instance, may be analysed in two dimensions. There is a relevant difference between the geometry of orientations in two and three dimensions. For instance, if one wishes to analyse different orientations in two dimensions, it is straight-forward to define a uniform distribution: it suffices to separate each direction by a constant angle.

Mathematically, the unitary circumference (or sphere in $\mathbb{R}^2$) is a one-dimensional space defined as $\mathbb{S}^1 = \{\mathbf{x} \in \mathbb{R}^2 \mid ||\mathbf{x}||^2 = 1\}$. More explicitly, $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^\mathsf{T}$ and $||\mathbf{x}||^2 = x^2 + y^2$. To parametrise an element in $\mathbb{S}^1$, it is common to define a mapping $\mathrm{S}^1 : \mathbb{R} \to \mathbb{S}^1$, *i.e.*, that maps a one-dimensional Euclidean space region into points of the circumference. Such map is given by $q \mapsto (x, y)$ and defined as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(q) \\ \sin(q) \end{bmatrix}. \tag{4.1}$$

An N-point uniform distribution on $\mathbb{S}^1$ is given by $\{q_i\} = \left(\frac{i-1}{N}\right) 2\pi$, for $i = 1, \dots, N$.

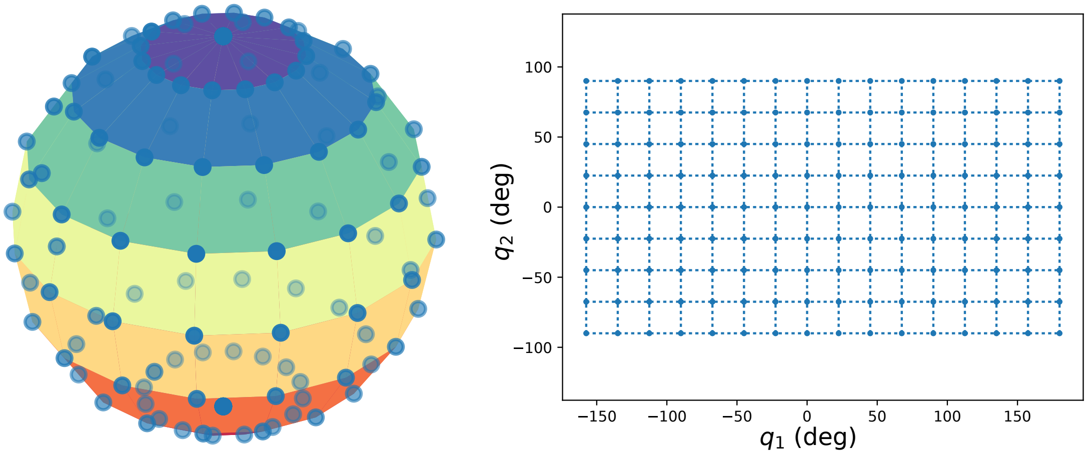
4.1.2 SPHERE PARAMETRISATIONS

The unitary sphere in $\mathbb{R}^3$ is a two-dimensional space defined as $\mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^3 \mid ||\mathbf{x}||^2 = 1\}$. More explicitly, $\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$ and $||\mathbf{x}||^2 = x^2 + y^2 + z^2$. A common way to parametrise an element in $\mathbb{S}^2$ is by defining a mapping $\mathrm{S}^2 : \mathbb{R}^2 \to \mathbb{S}^2$, *i.e.*, that maps a two-dimensional Euclidean space region into points of the three-dimensional sphere. Two classical parametrisations are discussed here: S^2_s, using spherical coordinates (or Euler angles) and, S^2_c, using cylindrical coordinates.

Using S_s^2, one has a map $(q_1, q_2) \mapsto (x, y, z)$, defined by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(q_1)\cos(q_2) \\ \sin(q_1)\cos(q_2) \\ \sin(q_2) \end{bmatrix}. \tag{4.2}$$

To cover the complete sphere, it is enough to define a region such that $q_1 \in (-\pi, \pi]$ and $q_2 \in [-\pi/2, \pi/2]$. Figure 4.1 illustrates how a (points) lattice in $\mathbb{R}^2$ maps into $\mathbb{S}^2$. Also, using



(a) (b)

Figure 4.1. Spherical coordinates parametrisation: (a) 3D geometry and (b) 2D parametrisation

S_c^2, one has a map $(q_1, q_2) \mapsto (x, y, z)$, defined by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(q_1)\sqrt{1-q_2^2} \\ \sin(q_1)\sqrt{1-q_2^2} \\ q_2 \end{bmatrix}. \tag{4.3}$$

To cover the complete sphere, it is enough to define a region such that $q_1 \in (-\pi, \pi]$ and $q_2 \in [-1, 1]$. Figure 4.2 illustrates how a lattice in $\mathbb{R}^2$ maps into $\mathbb{S}^2$.

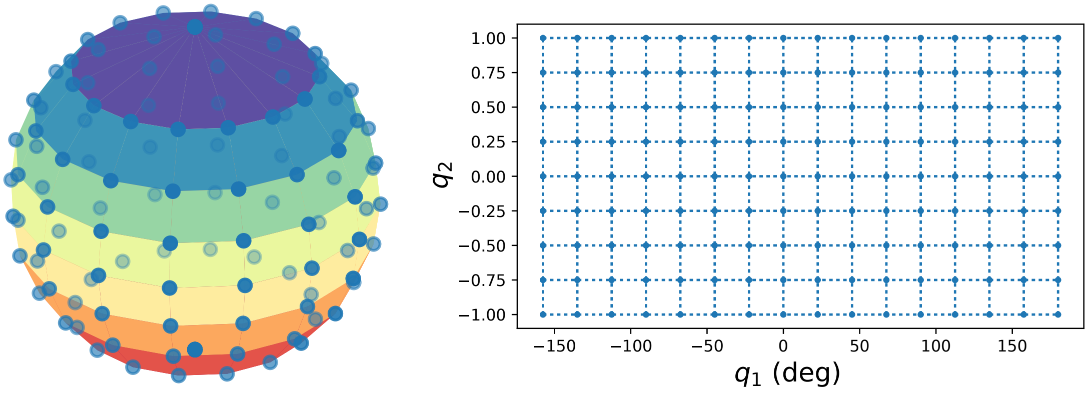
A difference between the two options is the way they map areas in $\mathbb{R}^2$ into $\mathbb{S}^2$. This can be qualitatively checked by comparing Figs. 4.1 (a) and 4.2 (a). More concretely, for S_s^2, one has that the tangent space is given by basis vectors

$$\mathbf{t}_{q_1} = \begin{bmatrix} -\sin(q_1)\cos(q_2) & \cos(q_1)\cos(q_2) & 0 \end{bmatrix}^{\mathsf{T}}$$

and

$$\mathbf{t}_{q_2} = \begin{bmatrix} -\cos(q_1)\sin(q_2) & -\sin(q_1)\sin(q_2) & \cos(q_2) \end{bmatrix}^{\mathsf{T}}.$$

In addition, one has $||\mathbf{t}_{q_1} \times \mathbf{t}_{q_2}|| = |\cos(q_2)|$, meaning that for a (q_1, q_2) the area differential is $\mathrm{d}A = |\cos(q_2)|\mathrm{d}q_1\mathrm{d}q_2$. This results shows how at different q_2 the area gets distorted when



(a) (b)

Figure 4.2. Cylindrical coordinates parametrisation: (a) 3D geometry and (b) 2D parametrisation

mapping from $\mathbb{R}^2$ to $\mathbb{S}^2$. This explains why, when using this parametrisation, areas get smaller near the poles, as Fig. 4.1 (a) illustrates. Additionally, note that $q_2 = \pm\pi/2$ are singular points. Subjectively speaking, this parametrisation does not deliver a 'fair' distribution of points in $\mathbb{S}^2$: points are sparse near the sphere's equator and close near the poles.

Conversely, for S^2_c, one has that the tangent space is given by basis vectors

$$\mathbf{t}_{q_1} = \begin{bmatrix} -\sin(q_1)\sqrt{1-q_2^2} & \cos(q_1)\sqrt{1-q_2^2} & 0 \end{bmatrix}^{\mathsf{T}}$$

and

$$\mathbf{t}_{q_2} = \begin{bmatrix} -\cos(q_1)\frac{q_2}{\sqrt{1-q_2^2}} & -\sin(q_1)\frac{q_2}{\sqrt{1-q_2^2}} & 1 \end{bmatrix}^{\mathsf{T}}.$$

In this case, one has $||\mathbf{t}_{q_1} \times \mathbf{t}_{q_2}|| = 1$, meaning that for a (q_1, q_2) the area differential is $\mathrm{d}A = \mathrm{d}q_1\mathrm{d}q_2$. This particular parametrisation does not distort area differentials between $\mathbb{R}^2$ and $\mathbb{S}^2$, as it is fairly apparent from Fig. 4.2 (a). Thus, area-wise, this last parametrisation allows one to better translate uniform point distributions in $\mathbb{R}^2$ to $\mathbb{S}^2$. Note that $q_2 = \pm 1$ are singular points. Subjectively speaking, this parametrisation gives a 'fairer' distribution of points in $\mathbb{S}^2$ from a regular distribution of points in $\mathbb{R}^2$.

4.1.3 UNIFORM POINT DISTRIBUTIONS ON A SPHERE

The concept of uniform N-point distribution in $\mathbb{S}^2$ is not trivial, because there is not an absolute criterion of uniformity for arbitrary N. The Platonic solids give an analytical solution for some $N \leq 20$; nonetheless, for larger N the solution is not unique. This subject has been approached for a while and numerous solutions exist [60, 109, 17]. Two classical cases are the stable carbon-60 molecules ($N = 60$) where atoms are arranged in a spherical football pattern [109] and Tammes' sphere packing problem, who studied the distribution of pores on pollen grains [60].

There are many criteria for measuring distribution uniformity. Two basic approaches may be classified as distance- and energy-based. The former evaluates point distributions based on the relative distance between all possible pairs of points; the latter evaluates a distribution based on measuring total potential energy as if each point were a charged particle. Focusing on distance-based criteria, it is necessary to use, *e.g.*, the Euclidean norm or any version of a cosine norm. For arbitrary points $\mathbf{x}_1$ and $\mathbf{x}_2$, the Euclidean distance is

$$\mathrm{d}_e(\mathbf{x}_1, \mathbf{x}_2) = ||\mathbf{x}_2 - \mathbf{x}_1||.$$

The cosine distance is

$$\mathrm{d}_c(\mathbf{x}_1, \mathbf{x}_2) = 1 - \frac{\mathbf{x}_1^{\intercal}\mathbf{x}_2}{||\mathbf{x}_1||\,||\mathbf{x}_2||}.$$

Furthermore, when points are in $\mathbb{S}^2$, the cosine norm becomes $\mathrm{d}_c(\mathbf{x}_1, \mathbf{x}_2) = 1 - \mathbf{x}_1^{\intercal}\mathbf{x}_2$. This last norm is consistent with a spherical geometry, because the dot product is closely related to the angle between two vectors and such angle is related to a distance measured over an arch. Because of this, often this distance is represented by the separation angle given by

$$\theta_{\mathrm{sep}} = \cos^{-1}\left(\mathbf{x}_1^{\intercal}\mathbf{x}_2\right) = \cos^{-1}\left(1 - \mathrm{d}_c(\mathbf{x}_1, \mathbf{x}_2)\right).$$

Distance-based distributions have two classical approaches that are mutually dual: *packing* and *covering* problems. Conceptually, these problems are not exclusive to spherical geometry and can be formulated on a Euclidean space as well. An N-point distribution is given by the finite set of points $\mathbf{X}^{(N)} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_i$ for $i = 1, \ldots, N$ are points in $\mathbb{S}^2$.

The *packing problem* proposes computing the maximin-distance criterion $\delta\left(\mathbf{X}^{(N)}\right)$, which is the minimum distance among all point pairings. More explicitly, it is given by [17]

$$\delta\left(\mathbf{X}^{(N)}\right) := \min_{1 \le i < j \le N} \mathrm{d}_{(\cdot)}(\mathbf{x}_i, \mathbf{x}_j), \tag{4.4}$$

where $\mathrm{d}_{(\cdot)}$ is any of the aforementioned distance measures. Therefore, the best packing is the one that maximises $\delta\left(\mathbf{X}^{(N)}\right)$, *i.e.*,

$$\mathbf{X}_*^{(N)} = \arg\max \delta\left(\mathbf{X}^{(N)}\right).$$

Figure 4.3 (a) shows a solution to the packing problem for $N = 116$, taken from N. Sloane's web-page [117]. The optimal maximin distance, namely $\delta^* = \delta\left(\mathbf{X}_*^{(N)}\right)$, has a geometrical meaning: $\delta^*/2$ may be used as radius to draw circular caps with centres on all $\mathbf{x}_i$; these caps never overlap and at most become tangent.

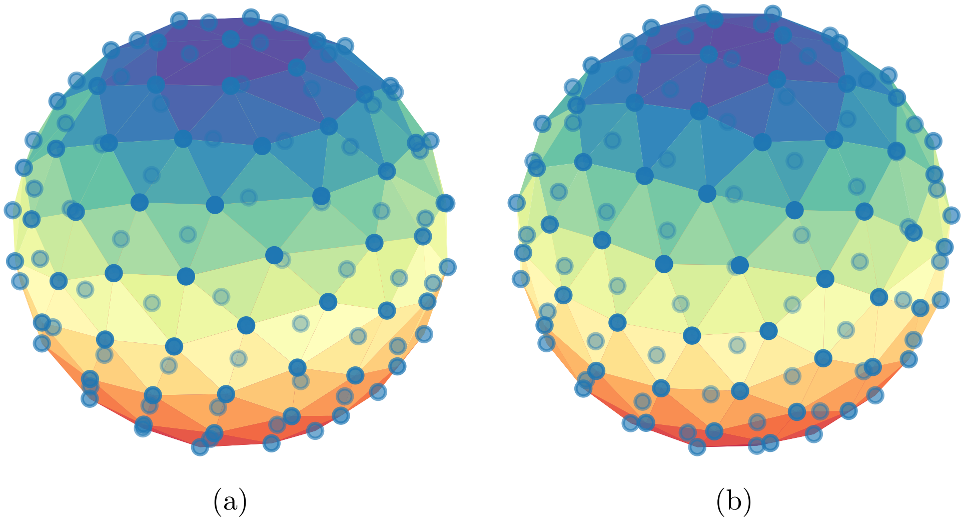


Figure 4.3. Uniform distribution of points on a sphere for $N = 116$: (a) packing and (b) covering. The data used to construct these plots were obtained from N. Sloane's web-page: http://neilsloane.com [117, 115]

The *covering problem* proposes computing the minimax-distance criterion $\rho\left(\mathbf{X}^{(N)}\right)$. Computing this criterion for distribution $\mathbf{X}^{(N)}$ requires exploring the distance between all $\mathbf{x} \in \mathbb{S}^2$, such that $\mathbf{x} \neq \mathbf{x}_i$, and its closest $\mathbf{x}_i$ and, finally, choosing the largest among all computed distances. Mathematically, this is

$$\rho\left(\mathbf{X}^{(N)}\right) := \max_{\mathbf{x}\in\mathbb{S}^2, \mathbf{x}\neq\mathbf{x}_i} \min_{1\leq i\leq N} \mathrm{d}_{(\cdot)}(\mathbf{x}_i, \mathbf{x}), \tag{4.5}$$

From this criterion, the best covering is the one that minimises $\rho\left(\mathbf{X}^{(N)}\right)$, *i.e.*,

$$\mathbf{X}_*^{(N)} = \arg\min \rho\left(\mathbf{X}^{(N)}\right).$$

Figure 4.3 (b) shows a solution to the covering problem for $N = 116$ taken from N. Sloane's web-page [115]. The optimal minimax distance, namely $\rho^* = \rho\left(\mathbf{X}_*^{(N)}\right)$, has a geometrical meaning: ρ^* may be used as radius to draw circular caps with centres on all $\mathbf{x}_i$; these caps cover all the sphere's surface.

4.2 GRAPHICAL REPRESENTATIONS

The graphical representations aim at answering the question of how points in the unitary sphere get mapped into some other surface by functions that appear on ROV motion models.

Mathematically, this means that the functions are manipulated so their domain gets restricted to the unitary sphere; then, the effect of the mapping is depicted as some transformation of the sphere. To do this, consider the graph $(r, \mathbf{x})$ defined by $r = \mathrm{f}(\mathbf{x})$, where r is a scalar, $\mathrm{f} : \mathbb{S}^2 \to \mathbb{R}$, $\mathbf{x} \in \mathbb{S}^2$, and $\mathbf{x} = \begin{bmatrix} e_x & e_y & e_z \end{bmatrix}^\intercal$. This graph can be depicted in three dimensions by the set $x = re_x$, $y = re_y$, and $z = re_z$ and, for instance, whenever r is constant, the region is a sphere with radius r.

Two types of graphical representations are discussed: those related to the mass matrix and manoeuvring meta-models. The former is primarily intended as a introduction to the idea of plotting graphs related to model functions. The latter is considered important in this work because it is aimed at illustrating the ability of the meta-model's basis functions to represent the geometry of more complex models.

4.2.1 INERTIA VISUALISATION

As mentioned in Chapter 2, the matrix of inertia (as defined by Eq. (2.6)) is given by

$$\mathbf{M}_{RB} = \left[\begin{array}{c|c} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \hline \mathbf{M}_{21} & \mathbf{M}_{22} \end{array}\right],$$

where $\mathbf{M}_{11} = m\mathbf{I}_{3\times3}$ is the mass matrix,

$$\mathbf{M}_{22} = \mathbf{I}_b = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix}$$

is the inertia tensor measured with respect to CO, and

$$\mathbf{M}_{21} = \mathbf{M}_{12}^\intercal = m\mathrm{S}(\mathbf{r}_g^b) = \begin{bmatrix} 0 & -mz_g & my_g \\ mz_g & 0 & -mx_g \\ -my_g & mx_g & 0 \end{bmatrix}.$$

Matrix $\mathbf{M}_{RB}$ takes part in model (2.5) through the rigid-body inertia term

$$\mathbf{M}_{RB}\dot{\nu} = \begin{bmatrix} \mathbf{M}_{11}\nu_1 + \mathbf{M}_{12}\nu_2 \\ \mathbf{M}_{21}\nu_1 + \mathbf{M}_{22}\nu_2 \end{bmatrix}.$$

As one could check, embedded in this term are functions $\mathrm{f}_{11} = \mathbf{M}_{11}\nu_1$, $\mathrm{f}_{12} = \mathbf{M}_{12}\nu_2$, $\mathrm{f}_{12} = \mathbf{M}_{21}\nu_1$, and $\mathrm{f}_{22} = \mathbf{M}_{22}\nu_2$. More specifically, f_{11} explains inertia force due to linear acceleration, f_{12} inertia force due to angular acceleration, f_{21} inertia moment due to linear acceleration, and f_{22} inertia moment due to angular acceleration. (Functions f_{12} and f_{21} are not to be confused with Coriolis effects.)

A representation for these functions may be done by assuming that velocities lie on the unitary sphere, *i.e.*, considering only the velocity direction by defining $\mathbf{e}_1 = \nu_1/||\nu_1||$ and $\mathbf{e}_2 = \nu_2/||\nu_2||$. According, to this, the functions can be further normalised to define new functions

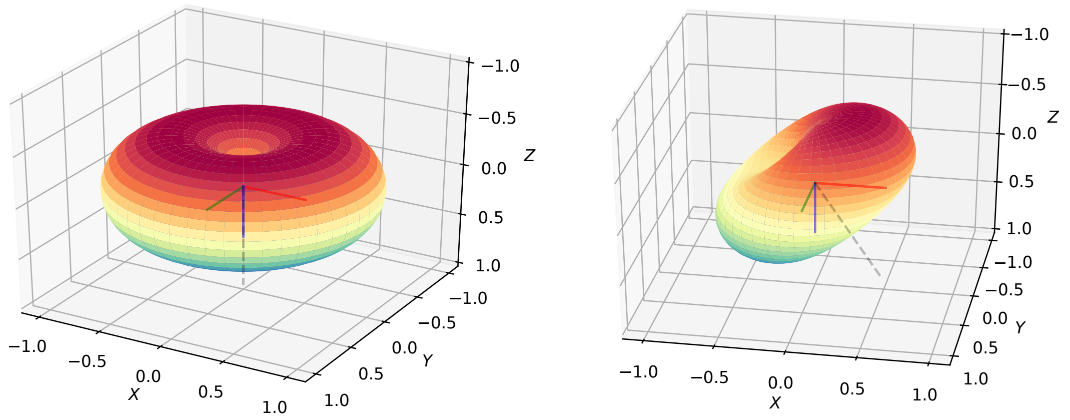
$$\begin{aligned}
\hat{\mathrm{f}}_{11} &= \frac{1}{m}\mathbf{M}_{11}\mathbf{e}_1 = \mathbf{I}_{3\times 3}\mathbf{e}_1,\\
\hat{\mathrm{f}}_{12} &= \frac{1}{||m\mathbf{r}_g^b||}\mathbf{M}_{12}\mathbf{e}_2,\\
\hat{\mathrm{f}}_{21} &= \frac{1}{||m\mathbf{r}_g^b||}\mathbf{M}_{21}\mathbf{e}_1,\\
\hat{\mathrm{f}}_{22} &= \frac{1}{\mathrm{trace}(\mathbf{M}_{22})}\left(\mathbf{M}_{22}\right)\mathbf{e}_2.
\end{aligned}$$

These functions give way to radii defined from each function's magnitude, *i.e.*, $r_{(\cdot,\cdot)} = ||\hat{\mathrm{f}}_{(\cdot,\cdot)}||$. For instance, matrix $\mathbf{M}_{11}$, related to mass, can be represented by radius $r_{11} = 1$, meaning that the effect on inertial force due to linear acceleration is equal at all directions.

Matrix $\mathbf{M}_{12}$ can be represented by radius

$$r_{12} = \sqrt{(e_{g_y}e_z - e_{g_z}e_y)^2 + (e_{g_z}e_x - e_{g_x}e_z)^2 + (e_{g_x}e_y - e_{g_y}e_x)^2},$$

where $\mathbf{e}_g^b = \mathbf{r}_g^b/||\mathbf{r}_g^b|| = \begin{bmatrix} e_{g_x} & e_{g_y} & e_{g_z} \end{bmatrix}^\mathsf{T}$ is the unitary vector of the centre of mass position. The resultant geometry is illustrated in Fig. 4.4. In Fig. 4.4 (a), the case where the centre of mass coordinates are in the z-axis direction is shown. Please note how the surface collapses to zero around the z-axis direction. This means that when accelerating in the z-axis direction there is no inertial force due to angular accelerations. Conversely, the radius goes to its maximum along directions in the xy-plane, meaning that when accelerating in the xy-plane the inertial forces due to angular accelerations is maximum. In Fig. 4.4 (b), a case where



(a)

(b)

Figure 4.4. Representation of mass matrix off-diagonal blocks: $\mathbf{M}_{12}$, $\mathbf{M}_{21}$

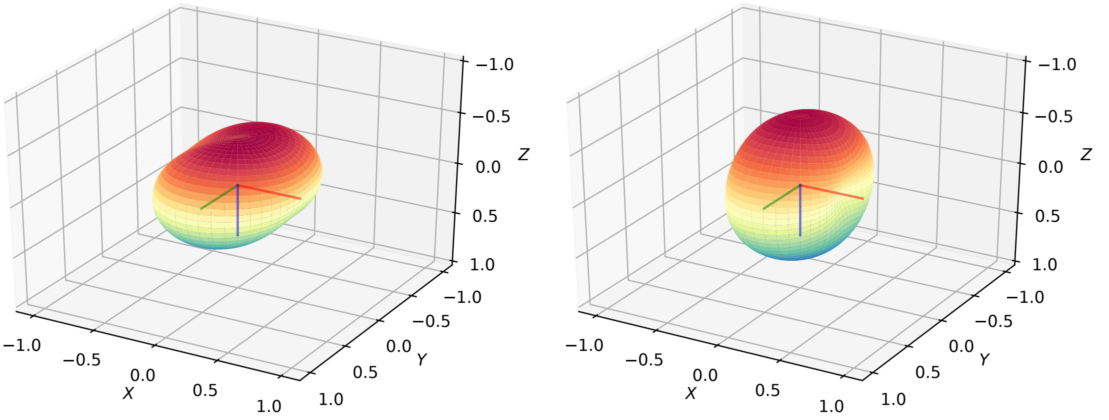
the centre of mass coordinates are in the xz-plane is shown. Please note how the surface is essentially the same but rotated: the surface's axis remains coincident with the centre of mass coordinates direction.

Additionally, please note that for matrix $\mathbf{M}_{21}$, radius $r_{21} = r_{12}$. This means that the inertial moment due to linear acceleration follows an analogous pattern to matrix $\mathbf{M}_{12}$'s. For example, when there is yaw acceleration there is no inertial moment due to linear accelerations. Conversely, when there are roll and/or pitch accelerations the inertial moments due to linear accelerations is maximum.

Matrix $\mathbf{M}_{22}$ can be represented by radius

$$r_{22} = \sqrt{(k_{xx}e_x + k_{xy}e_y + k_{xz}e_z)^2 + (k_{xy}e_x + k_{yy}e_y + k_{yz}e_z)^2 + (k_{xz}e_x + k_{yz}e_y + k_{zz}e_z)^2},$$

where the $k_{(\cdot,\cdot)}$'s are trace-normalised radii of gyration. The resultant geometries related to PIONERO500 and MINERVA are illustrated in Fig. 4.5. The main difference between the two ROV's models lies in that MINERVA's inertia matrix is diagonal, *i.e.*,



(a)

(b)

Figure 4.5. Representation of inertia matrix $\mathbf{M}_{22}$: (a) PIONERO500's and (b) MINERVA's

$$r_{22} = \sqrt{(k_{xx}e_x)^2 + (k_{yy}e_y)^2 + (k_{zz}e_z)^2},$$

and in that of PIONERO500 some products of inertia are non-zero. Please note how, comparatively, PIONERO500's graph is less symmetric. Note also that the radius is larger around the directions with greater rotational inertia.

4.2.2 POLYNOMIALS FOR META-MODELLING

Manoeuvring meta-model (2.13), a model for hydrodynamic forces and moments, is often written in component form, *i.e.*, $\tau_H = \begin{bmatrix} X & Y & Z & K & M & N \end{bmatrix}^\intercal$, a vector with three force and three moment components. Additionally, each component is approximated by using some subspace of functions, defined by some basis. For instance, for load X one has

$$X = \sum_i X_i \, \mathrm{g}_{X_i}(\nu_r, \dot{\nu}_r),$$

where the X_i are coefficients and g_{X_i} are the basis functions. Often ν_r and $\dot{\nu}_r$ are written in their components as well, where, dropping the sub-index for convenience, one writes $\nu_r = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^\intercal$ and $\dot{\nu}_r = \begin{bmatrix} \dot{u} & \dot{v} & \dot{w} & \dot{p} & \dot{q} & \dot{r} \end{bmatrix}^\intercal$.

A common basis for X load due to drag is given by $f_X = \{u, u|u|, u|v|, u|w|\}$, that includes a linear term and quadratic terms in modulo form. This basis gives way to meta-models such as

$$X(u, v, w) = X_u u + X_{u|u|} u|u| + X_{u|v|} u|v| + X_{u|w|} u|w|.$$

In these meta-models one might consider various spaces: input space, *e.g.*, that of (u, v, w); output space, *e.g.*, that of X; and coefficient coordinates space, *e.g.*, that of

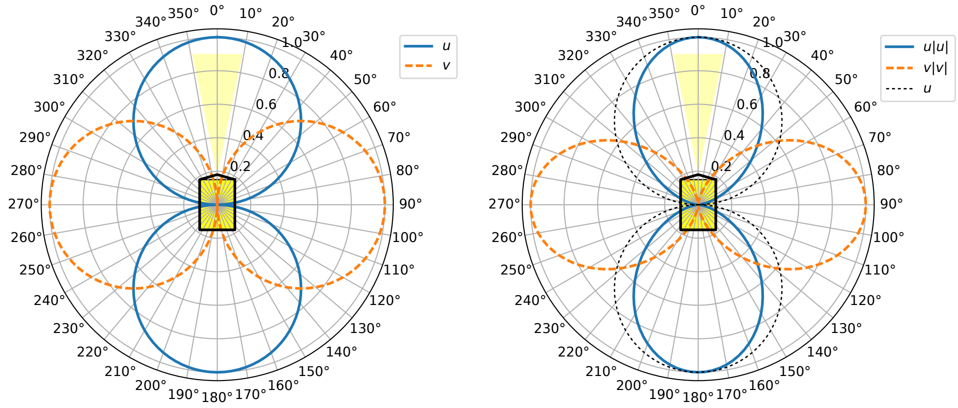
$$(X_u, X_{u|u|}, X_{u|v|}, X_{u|w|}).$$

This is to say that, considering that the output space is a transformation of the input space, the reachable output space results from finding proper coefficient coordinates in the directions given by the functions basis. This means that a good meta-model depends on properly choosing basis functions and computing coefficients. Here, the problem of visualising the basis functions is considered.

To develop visualisations related to the basis functions, the input space is restricted to the unitary sphere, *i.e.*, $u^2 + v^2 + w^2 = 1$ and the output space is represented by $x = ru$, $y = rv$, and $x = rw$, where $r \geq 0$ is the magnitude of each basis function. For instance, for basis function u, the visual representation is given by $x = |u|u$, $y = |u|v$, and $x = |u|w$, and for basis function $u|u|$, the visual representation is given by $x = u^2 u$, $y = u^2 v$, and $x = u^2 w$ (redundancy $u^2 u$ is intended).

As a first approach, two-dimensional representations are considered. Figure 4.6 (a) depicts the space of linear basis functions u and v as the (x, y) graph given by $x = |u|u$ and $y = |u|v$ for basis function u. Conversely, the graph given by $x = |v|u$ and $y = |v|v$ depicts the space of basis function v. Furthermore, Fig. 4.6 (b) depicts the space of quadratic basis functions

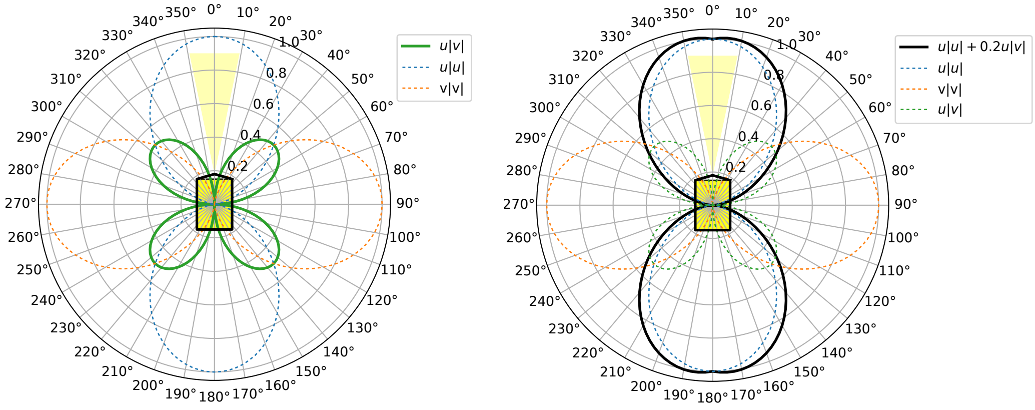
$|u|u$ and $|v|v$ as the graph $x = u^2u$ and $y = u^2v$ for basis function $|u|u$ and the graph $x = v^2u$ and $y = v^2v$ for basis function $|v|v$.



(a) (b)

Figure 4.6. Representation of basis functions in two-dimensions, part 1: (a) u and v and (b) $u|u|$ and $v|v|$. Please note the dotted line in (b) that allows comparing linear to quadratic functions

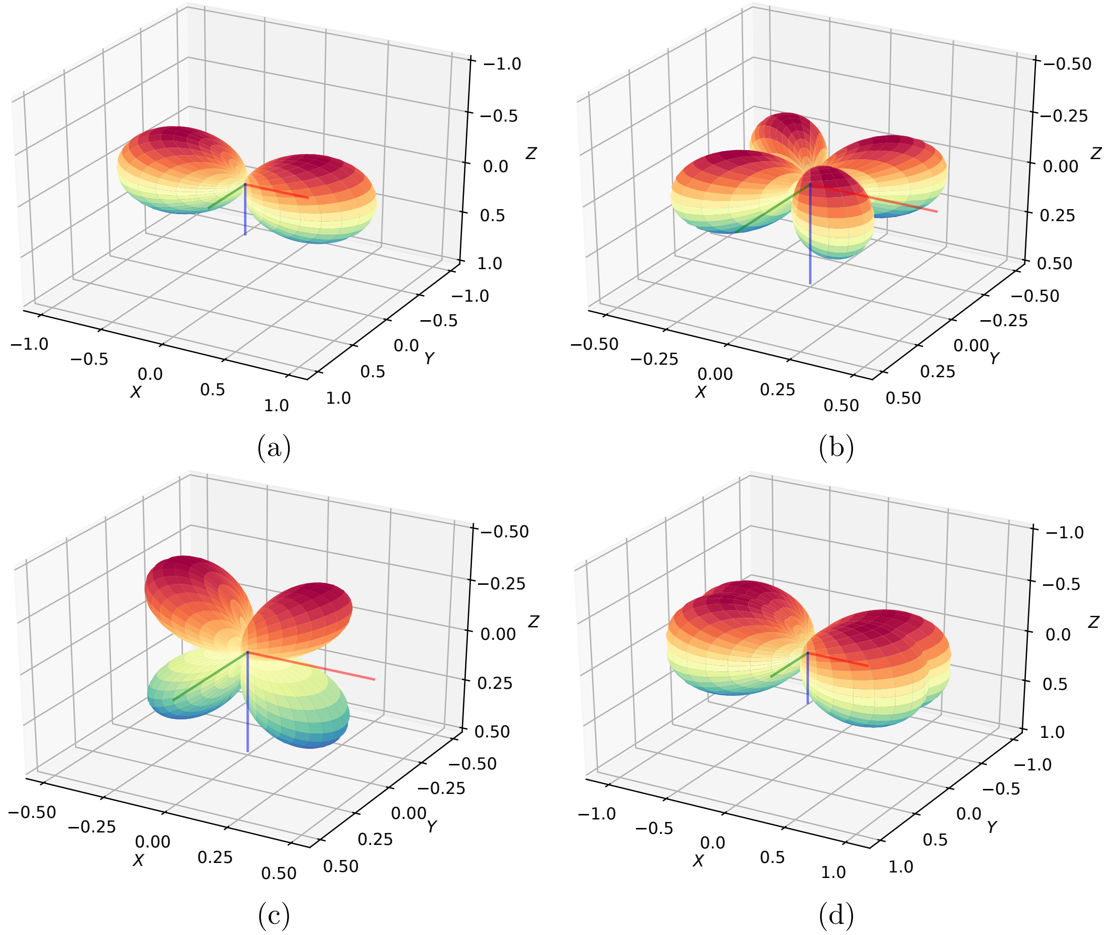
Moreover, Fig. 4.7 (a) depicts the space of cross-coupled basis function $|v|u$ as the graph $x = |uv|u$ and $y = |uv|v$. In the figure, it is compared to the quadratic basis functions of Fig. 4.6 (b). The effect of considering a cross-coupling term is illustrated in Fig. 4.7 (b), where the solid (black) line represents the output space given by some linear combination of quadratic and cross-coupling basis functions, *i.e.*, $|u|u$ and $|v|u$. Please note how the space it is able to represent is richer.



(a) (b)

Figure 4.7. Representation of basis functions in two dimensions: part 2

Three-dimensional representations are considered next. Figure 4.8 (a) depicts the space of quadratic basis function $|u|u$ as the (x, y, z) graph given by $x = u^2u$, $y = u^2v$, and $z = u^2w$. This is the same function as in Fig. 4.6 (b) but extended to the z-axis. Cross-coupling functions are considered in Figs. 4.8 (b) and (c). Figure 4.8 (b) depicts the space of basis function $|v|u$ as the graph given by $x = |uv|u$, $y = |uv|v$, and $z = |uv|w$. This is the same function as in Fig. 4.7 (a) but extended to the z-axis. Conversely, Fig. 4.8 (c) depicts the space of basis function $|w|u$ as the graph given by $x = |uw|u$, $y = |uw|v$, and $z = |uw|w$. Finally, Fig. 4.8 (c) illustrates how a linear combination of basis functions $|u|u$, $|v|u$, and $|w|u$ looks like.



(a)

(b)

(c)

(d)

Figure 4.8. Representation of basis functions in three dimensions

4.3 INTEGRATION

Consider the computation of area integrals over $\mathbb{S}^2$, *i.e.*, integrals of functions of $\mathbf{x} \in \mathbb{S}^2$ where $\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$. Such integrals are in general

$$\int_S \mathrm{f}(x, y, z)\mathrm{d}S,$$

where $\mathrm{d}S$ is a properly defined area differential.

4.3.1 SECOND MOMENT INTEGRALS

Considering functions of the type used on meta-models' bases, *e.g.*, $\{x, x|x|, x|y|, x|z|\}$, one can argue that some important integrals are those of the second order moments, *i.e.*, of the functions and their combinations over the whole domain. This is called the basis function moment matrix and its components may be computed as

$$M_{ij} = \int_S \mathrm{f}_i\, \mathrm{f}_j\, \mathrm{d}S,$$

where i and j are indices on the basis list.

For instance, for the previously defined list, some possible integrals are, *e.g.*, $M_{11} = \int_S x^2 \mathrm{d}S$, $M_{22} = \int_S x^4 \mathrm{d}S$, and $M_{23} = \int_S |x|^3|y|\mathrm{d}S$. These integrals can be computed analytically and a summary of these is shown in Table 4.1. These integrals represent the variance and covariance related to the functions basis.

Table 4.1. Basis functions second moment integrals

$\int_S \mathrm{f}_i\, \mathrm{f}_j\, \mathrm{d}S$	x	$x\|x\|$	$x\|y\|$	$x\|z\|$
x	$4\pi/3$	symm	symm	symm
$x\|x\|$	π	$4\pi/5$	symm	symm
$x\|y\|$	$\pi/2$	$16/15$	$4\pi/15$	symm
$x\|z\|$	$\pi/2$	$16/15$	$8/15$	$4\pi/15$

4.3.2 APPROXIMATIONS BY LEBESGUE INTEGRATION

The aforementioned integrals may be approximated in a practical manner by using the idea of *Lebesgue integrals*. In this case, the approximations depend on having a uniform distribution of N points in the sphere, so that all points have equal probability of occurring. The Lebesgue integral is written as

$$\int_S \mathrm{f}(x, y, z) \mathrm{d}\mu(x, y, z),$$

where $\mu(x, y, z)$ is the area measure at each specified point. For a sphere, the area measure satisfies $\int_S \mathrm{d}\mu(x, y, z) = 4\pi$. Then, because equal probability is assigned to each point, *i.e.*, each point is inside a subset with measure $4\pi/N$, the integral may be approximated as

$$\int_S \mathrm{f}(x, y, z) \mathrm{d}\mu(x, y, z) = \frac{4\pi}{N} \sum_{i=1}^{N} \mathrm{f}(x_i, y_i, z_i),$$

which is a simpler computation. A numerical comparison between the analytical solution and Lebesgue integral approximation is shown in Table 4.2, where covering and packing uniform distributions of $N = 116$ points, as shown in Figs. 4.3 (a) and (b), were used.

Table 4.2. Approximation of integrals using Lebesgue integration on $N = 116$ points

Term	Analytical	L-Covering	L-Packing
x-x	4.1887902	4.17945531	4.18999603
$x\|x\|$-x	3.14159265	3.13307049	3.14704616
$x\|y\|$-x	1.57079633	1.55716479	1.56813755
$x\|z\|$-x	1.57079633	1.56069676	1.5736585
$x\|x\|$-$x\|x\|$	2.51327412	2.50593996	2.52024737
$x\|y\|$-$x\|x\|$	1.06666667	1.05297611	1.06642479
$x\|z\|$-$x\|x\|$	1.06666667	1.05744632	1.06946316
$x\|y\|$-$x\|y\|$	0.83775804	0.83748503	0.82985865
$x\|z\|$-$x\|y\|$	0.53333333	0.52770889	0.53838156
$x\|z\|$-$x\|z\|$	0.83775804	0.83603031	0.83989001

Chapter 5

Design of computer experiments

In the context of this thesis, any computation scenario is a *computer experiment.* Consequently, any selection of computation scenarios is a computer experiment *design.* It is rather common to select different computation scenarios heuristically, specially because myriad computations, like time-domain simulations, run in the order of seconds or milliseconds. Nonetheless, a heuristic approach to define scenarios may be inconvenient when dealing with resource-demanding computations, such as those of computational fluid dynamics (CFD). In consequence, *heuristic designs* will refer to choosing the scenarios without using any formal procedure or criterion. Conversely, to use more systematic alternatives, this chapter introduces methods to formally organise computer experiments (or designs) that will be useful in the topics to come.

5.1 TECHNIQUES FOR THE DESIGN OF COMPUTER EXPERIMENTS

Design of computer experiments (DoCE) is a well-established discipline [44]. In [75, 50] one can find a broad review on the subject and in [44] a comprehensive reference textbook. The contents of this chapter are based on [60, 109, 44, 75, 96, 17, 50]; however, some fundamental concepts are presented here rather hastily. As mentioned, computer experiments can be understood as sets of simulation scenarios. In the language of DoE, those sets have *factors*, *i.e.*, qualitative or quantitative variables of the computational model that could be modified at will. For example, inflow velocity could be a quantitative factor and turbulence model a qualitative one. Herein, factors are also called *inputs*. Geometrically, factors define a domain or input variable space, or simply *input space*. For instance, if inflow direction is the input, then Figs. 4.1, 4.2, and 4.3 of Chapter 4 illustrate an input space. A point in the input space defines a particular computation or *experimental point*. A *run* or *trial* is a computation over an experimental point: this constitutes an important difference to 'real' experiments, because all trials over the same experimental point run give the same result. The result of a computation is called the *response*. A *design table* or *design* is a list of experimental points that somehow fill the input space.

Roughly speaking, DoCE can be classified as shown in Fig. 5.1. DoCE methods could be static or adaptive. For static methods, the design table is computed once and then executed; adaptive methods update the design table depending on the responses. This work only considers static methods; thus, adaptive methods are out of scope. Subsequently, static methods can be model-free or model-based (the names system-free and system-aided are also used [50]). Model-free methods concentrate on properly filling the input space, without regarding the purpose of the experiments. These methods are convenient for exploratory analyses, when the experiments purpose is not clear, and, perhaps, for mixed-purpose cases. Model-based methods take advantage of knowing the purpose of the experiments beforehand: they go beyond filling an input space and further use the model and response spaces. These methods may consider, among others, regression, integration, and interpolation as purposes.

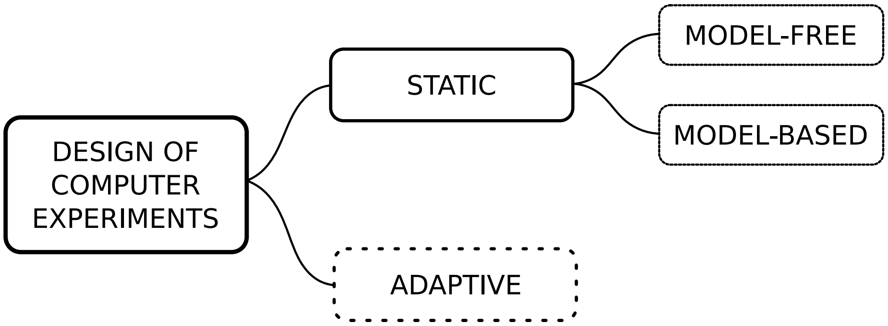


Figure 5.1. DoCE classification

It is important to mention that this work goes as well beyond the idea of widely used (full) factorial design. Simply put, a *full factorial design* discretises each of the s inputs (factors) in, let us say, n_j levels. Then, the design table is constructed by combining all possible level combinations: a total of $n_1 \times n_2 \times \ldots \times n_s$ runs. This type of design grows unmanageably when the number of inputs or levels is increased. For instance, for a 2-input design with 5 levels at each variable the number of combinations is 25, but with 10 levels at each variable, the number of combinations is 100. In cases such as viscous-flow computations, there is a considerable difference between performing 25 rather than 100 computations.

5.2 MODEL-FREE DESIGNS

Model-free DoCE methods are also called space-filling methods, because the purpose is to fill the input space properly. Supposing there are s input variables, the input space may be an s-dimensional hypercube (a hypercube region in $\mathbb{R}^s$). Without losing generality, this space could be restricted to the unitary hypercube $\mathbb{C}_1^s \subset \mathbb{R}^s$, where each variable lies in the range $[0, 1]$. Moreover, this work also considers input spaces such as $\mathbb{S}^2$ (please note that general spaces such as $\mathbb{S}^{n-1}$ are possible but not considered here). Space filling methods could be (pseudo) random or use a uniformity criterion to construct design tables. Among (pseudo) random methods, one can find Monte-Carlo Sampling (MCS) or Latin-Hypercube Sampling (LHS). Uniform designs may be deterministic and have a lattice design, such as the commonly used full-factorial design of DoE or equivalent, or have some level of randomness.

Hypercubic and spheric spaces are both relevant to this work. Some problems arrange naturally in one or other topology. Hypercubic spaces arise when variables can be defined by simple intervals and normalised to $[0, 1]$ (or equivalent). Here, spheric spaces ($\mathbb{S}^2$) arise when considering three-dimensional orientation as an input. $\mathbb{S}^1$ may be considered by defining an interval in $\mathbb{R}$, *i.e.*, one of some hypercube's dimensions. Also, it is possible to construct designs in $\mathbb{S}^2$ from designs in $\mathbb{R}^2$ by using, *e.g.*, parametrisation (4.3). The usage of this parametrisation is proposed here as a plausible practical approach. Furthermore, hypercube and spheric spaces are not mutually exclusive; on the contrary, they are considered complementary to each other.

In the methods to come the following definitions are used:

- The input space is given by s inputs in a unitary hypercube $\mathbb{C}_1^s$.
- The total number of runs is given by N.
- A design is given by $\mathbf{X}^{(N)} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where each $\mathbf{x}_i \in \mathbb{C}_1^s$ and

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} & x_{21} & \ldots & x_{is} \end{bmatrix}^\mathsf{T}.$$

5.2.1 A BRIEF NOTE ON MONTE-CARLO DESIGNS

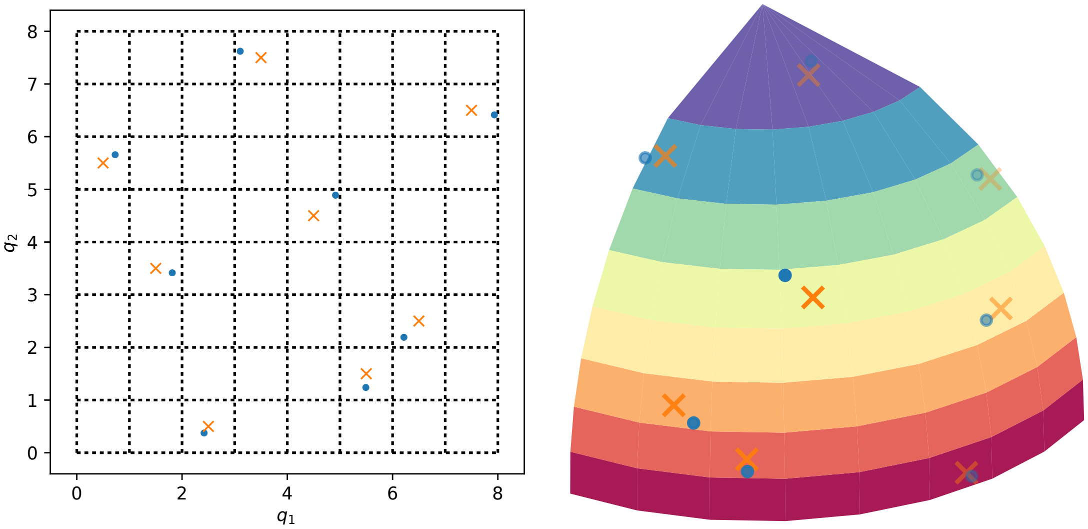
Monte-Carlo Design (MCD) is an option for construction of design tables. A MCD with N runs is constructed by selecting (pseudo) random values for each input at each run i. This type of design is straightforward to construct, suitable for exploratory analysis and where computations run rather fast.

5.2.2 LATIN HYPERCUBE DESIGN

A Latin Hypercube Design (LHD), *i.e.*, constructed by using LHS, is a widely used method for constructing design tables. A standard Latin Hypercube divides each hypercube's dimension equally. Supposing one wants an experiment with a total of N runs, each dimension is divided in N parts. In the design table, each row represents an experimental run and each column an input. Table 5.1 shows how the process may look like for a 2-input, 8-run design ($s = 2$, $N = 8$); this table facilitates explaining the process. The runs will be numbered from 1 to N, *i.e.*, from a list $\{1, 2, \ldots, N\}$. Then, to obtain each column representing an input, a random permutation of the list should be computed, this is Step 1 in Table 5.1. Then, to the permuted list a number between 0 and 1 should be substracted, either a random quantity (Step 2, random) or 0.5 (Step 2, mid-point). Finally, to match a unitary hypercube and compute a proper $\mathbf{X}^{(N)}$, the results must be normalised by N. In Fig. 5.2, this process is further illustrated in (a) $\mathbb{R}^2$ and (b) $\mathbb{S}^2$ spaces ($\mathbb{S}^2$ space was obtained using parametrisation (4.3) over an eighth sphere). As it becomes apparent in Fig. 5.2 and seeing the 2-D plot as rows and columns, this design organises points so that there is just one point for each row and column. As one may expect, each time a LHD is constructed a different result is obtained,

Table 5.1. LHD table

Run	Step 1		Step 2, random		Step 2, mid-point	
	q_1	q_2	q_1	q_2	q_1	q_2
1	7	3	6.215	2.191	6.5	2.5
2	3	1	2.419	0.375	2.5	0.5
3	8	7	7.934	6.414	7.5	6.5
4	6	2	5.492	1.239	5.5	1.5
5	5	5	4.916	4.889	4.5	4.5
6	2	4	1.814	3.416	1.5	3.5
7	1	6	0.729	5.657	0.5	5.5
8	4	8	3.106	7.621	3.5	7.5



(a) (b)

Figure 5.2. LHD example

and, therefore, some designs will fill the space better than others. Fig. 5.2 (b) shows how the design looks in $\mathbb{S}^2$, using parametrisation (4.3) over an eighth sphere.

5.2.3 ORTHOGONAL HYPERCUBE DESIGN

A drawback of LHS is that it could leave some sections on the space unexplored, this has been remarked by, *e.g.*, [44, 50]. An Orthogonal Hypercube Design (OHD) may be understood as a type of LHD that keeps a level of randomness but gives guarantees that all regions are explored. Here, a basic version of OHD is considered. First, define a number n. Then, the space is divided into n^s big chunks and, subsequently, each chunk in n^s smaller pieces. The experiment has a total of $N = n^s$ runs. The idea behind this OHD is that each big chunk gets one point that is randomly located in one of the smaller pieces, following the rules of LHD. As in LHD, each dimension gets divided in N parts.

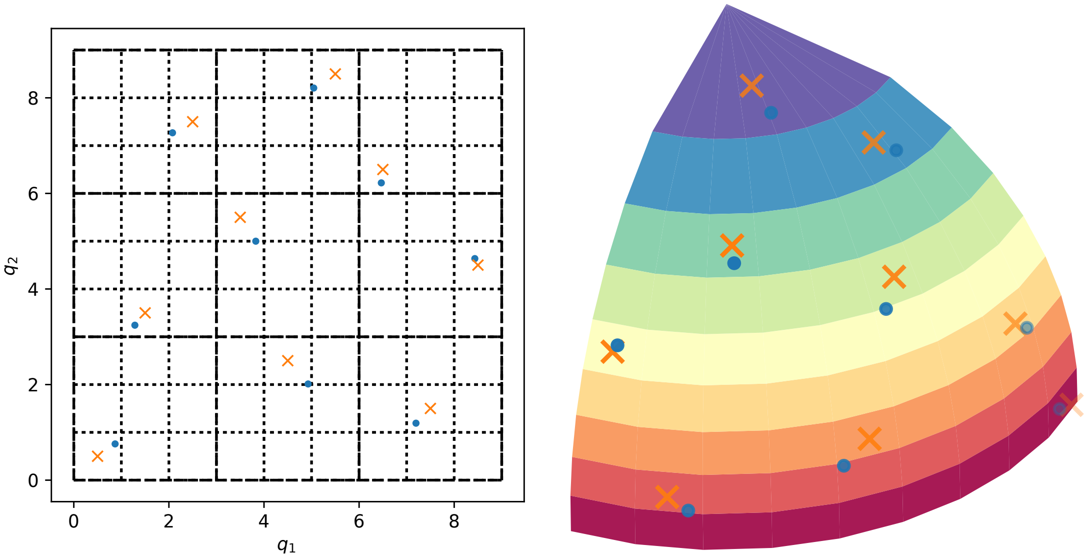
Table 5.2 shows how the process may look like for a 2-input problem where $n = 3$; hence, the number of runs is $N = 9$. From Step 1, it is apparent how the table has three big sections that are identified by repeated numbers from 1 to 3, written as $\{1, 1, 1, 2, 2, 2, 3, 3, 3\}$. Note how $1 \to \{1, 2, 3\}$, $2 \to \{4, 5, 6\}$, and $3 \to \{7, 8, 9\}$. Then, q_1 is obtained from random permutations on the same section. Afterwards, q_2 is constructed from a permutation of $\{1, 1, 1, 2, 2, 2, 3, 3, 3\}$, namely $\{1, 2, 3, 1, 2, 3, 1, 2, 3\}$. Note how q_2 is obtained from a random permutation that holds the rule $1 \to \{1, 2, 3\}$, $2 \to \{4, 5, 6\}$, and $3 \to \{7, 8, 9\}$. Finally, Step

Table 5.2. OHD table

Run	Step 1		Step 2, random		Step 2, mid-point	
	q_1	q_2	q_1	q_2	q_1	q_2
1	$1 \to 1$	$1 \to 1$	0.869	0.758	0.5	0.5
2	$1 \to 2$	$2 \to 4$	1.284	3.242	1.5	3.5
3	$1 \to 3$	$3 \to 8$	2.076	7.269	2.5	7.5
4	$2 \to 5$	$1 \to 3$	4.928	2.013	4.5	2.5
5	$2 \to 4$	$2 \to 6$	3.827	5.000	3.5	5.5
6	$2 \to 6$	$3 \to 9$	5.046	8.205	5.5	8.5
7	$3 \to 8$	$1 \to 2$	7.196	1.192	7.5	1.5
8	$3 \to 9$	$2 \to 5$	8.434	4.636	8.5	4.5
9	$3 \to 7$	$3 \to 7$	6.467	6.221	6.5	6.5

2 is the same as in LHD, where the final number is calculated in a mid-point or randomly. This design table requires normalisation as well.

Figure 5.3 illustrates this process in $\mathbb{R}^2$ (a) and $\mathbb{S}^2$ (b) spaces. Figure 5.3 (a) shows the design in $\mathbb{R}^2$. This plot shows the 9 big chunks used to equally divide the space, as well as the 9 equal parts that divide each chunk. Each big chunk has one experimental point located inside one small part, either at the centre or at a random location. Each dimension is divided in 9



(a)

(b)

Figure 5.3. OHD example

parts. As in LHD, this design organises points so that there is just one point for each row and column.

5.2.4 MEASURES OF UNIFORMITY

There are numerous criteria for assessing uniformity of a design. Three major categories are discrepancy-, distance-, and energy-based. All criteria allow one to compute an index to objectively identify the better of any two design table options. Among discrepancy-based criteria one can find star-, centred- and wrap-around-discrepancy. Among distance-based criteria one can find the aforementioned (in Sec. 4.1.3) minimax and maximin distances. And among energy-based criteria one can find Coulomb-, s-, and logarithmic-energy measures.

Measurements of *discrepancy* look forward to computing how well points are distributed in relation to regions of the input space. This is, a relation between an hyper-volume measure in $\mathbb{C}_1^s$ and the count of points that lie inside. For computing *star-discrepancy*, let us suppose that $\mathbf{x}$ is any point in the input space. Subsequently, this point defines a hyper-volume $[\mathbf{0}, \mathbf{x})$, the hypercube defined by vertices $\mathbf{0}$ and $\mathbf{x}$. This, consequently, defines a corresponding volume measure $\mathrm{Vol}\left([\mathbf{0}, \mathbf{x})\right)$ and a count of points of $\mathbf{X}^{(N)}$ that lie inside, defined as $\#\left(\mathbf{X}^{(N)}, [\mathbf{0}, \mathbf{x})\right)$. From these definitions, the star-discrepancy is computed by performing a search among all $\mathbf{x} \in \mathbb{C}_1^s$ and checking the largest difference between the fraction of points inside the hyper-volume and corresponding volume measure. Mathematically, this is

$$D_* \left(\mathbf{X}^{(N)}\right) = \max_{\mathbf{x} \in \mathbb{C}_1^s} \left| \frac{\#\left(\mathbf{X}^{(N)}, [\mathbf{0}, \mathbf{x})\right)}{N} - \mathrm{Vol}\left([\mathbf{0}, \mathbf{x})\right) \right| .$$

This discrepancy measure create regions with respect to the origin $\mathbf{0}$. Other measures of discrepancy have the same concept as star-discrepancy but differ in the way they create the regions.

Centred discrepancy is computed by defining the hyper-volume related to an arbitrary $\mathbf{x}$ using points $\mathbf{x}$ and its closest corner of $\mathbb{C}_1^s$. Mathematically, this is computed as follows

$$\begin{aligned} \left[D_C \left(\mathbf{X}^{(N)}\right)\right]^2 = \left(\frac{13}{12}\right)^s - \frac{2}{N} \sum_{k=1}^{N} \prod_{j=1}^{s} \left[1 + \frac{1}{2}|x_{kj} - 0.5| - \frac{1}{2}|x_{kj} - 0.5|^2\right] \\ + \frac{1}{N^2} \sum_{k=1}^{N} \sum_{j=1}^{N} \prod_{i=1}^{s} \left[1 + \frac{1}{2}|x_{ki} - 0.5| + \frac{1}{2}|x_{ji} - 0.5| - \frac{1}{2}|x_{ki} - x_{ji}|\right] . \end{aligned}$$

Wrap-around discrepancy follows the idea of not using any corner of $\mathbb{C}_1^s$ and mathematically is computed as

$$\left[D_W\left(\mathbf{X}^{(N)}\right)\right]^2 = -\left(\frac{4}{3}\right)^s + \frac{2}{N^2}\sum_{k,j=1}^{N}\prod_{j=1}^{s}\left[\frac{3}{2} + |x_{ki} - x_{ji}|(1 - |x_{ki} - x_{ji}|)\right].$$

Distance-based criteria are often computed by using minimax and maximin distances. These distances could be measured either in $\mathbb{R}^s$ or $\mathbb{S}^2$. For $\mathbb{S}^2$, these distances where defined by eqs. (4.4) and (4.5) as $\delta\left(\mathbf{X}^{(N)}\right)$ and $\rho\left(\mathbf{X}^{(N)}\right)$. For $\mathbb{R}^s$, the definitions are similar but using the Euclidean norm. *Maximin distance* is

$$\delta\left(\mathbf{X}^{(N)}\right) := \min_{1\leq i<j\leq N} \mathrm{d}_e(\mathbf{x}_i, \mathbf{x}_j)$$

and *minimax distance* is

$$\rho\left(\mathbf{X}^{(N)}\right) := \max_{\mathbf{x}\in\mathbb{R}^s, \mathbf{x}\neq\mathbf{x}_i}\ \min_{1\leq i\leq N} \mathrm{d}_e(\mathbf{x}_i, \mathbf{x}).$$

Energy-based criteria use Coulomb-, s-, and logarithmic-energy measures. *Coulomb energy* is given by

$$E_C\left(\mathbf{X}^{(N)}\right) = \sum_{k=1}^{N-1}\sum_{j=1}^{k}\frac{1}{||\mathbf{x}_j - \mathbf{x}_k||},$$

s-energy (2-energy being the most common) is given by

$$E_s\left(\mathbf{X}^{(N)}\right) = \sum_{k=1}^{N-1}\sum_{j=1}^{k}\frac{1}{||\mathbf{x}_j - \mathbf{x}_k||^{\mathrm{s}}},$$

and *log-energy* is given by

$$E_{\ln}\left(\mathbf{X}^{(N)}\right) = \sum_{k=1}^{N-1}\sum_{j=1}^{k}\ln\frac{1}{||\mathbf{x}_j - \mathbf{x}_k||},$$

5.3 DESIGNS FOR META-MODEL REGRESSION

When data from computer experiments are available, a linear least-squares regression may be used to find a meta-model. Here, recalling the example of load X from eq. (2.14), one has that a meta-model is

$$X = \sum_i X_i\, \mathrm{g}_{X_i}(\nu_r, \dot{\nu}_r).$$

To make the analysis general, assume that any load component $y \in \mathbb{R}$ is modelled exactly by

$$y = \mathrm{f}(\mathbf{x}),$$

where function $\mathrm{f} : \mathbb{R}^n \to \mathbb{R}$ defines the map $\mathbf{x} \mapsto y$. A meta-model is supposed to approximate f and make an estimate $\hat{y}$ of y. Such estimate is given by [130]

$$\hat{y} = \sum_{i=1}^{M} b_i \, \mathrm{g}_i(\mathbf{x}) = \left[\begin{array}{ccc} \mathrm{g}_1(\mathbf{x}) & \cdots & \mathrm{g}_M(\mathbf{x}) \end{array} \right] \left[\begin{array}{c} b_1 \\ \vdots \\ b_M \end{array} \right] = \mathbf{g}^\mathsf{T}(\mathbf{x})\mathbf{b} = \mathbf{b}^\mathsf{T}\mathbf{g}(\mathbf{x}), \tag{5.1}$$

where M is the function's basis dimension, b_i support coefficients, and g_i basis functions. Because the estimate is not exact, the residual function is given by

$$\mathrm{r}(\mathbf{x}) = y - \hat{y} = \mathrm{f}(\mathbf{x}) - \mathbf{g}^\mathsf{T}(\mathbf{x})\mathbf{b},$$

and the exact model may be written as

$$y = \hat{y} + \mathrm{r}(\mathbf{x}) = \mathbf{g}^\mathsf{T}(\mathbf{x})\mathbf{b} + \mathrm{r}(\mathbf{x}),$$

where $e = \mathrm{r}(\mathbf{x})$ represents the error at any $\mathbf{x}$. Here, $\mathrm{r}(\mathbf{x})$ is assumed a zero-mean, Gaussian normal distribution with variance σ^2.

To find the metamodel coefficients, let us suppose there is a set of N database entries, where for each $j = 1, \ldots, N$ one has $y^{(j)}$ and $\mathbf{x}^{(j)}$. This means that a design given by $\mathbf{X}^{(N)} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ was carried through. Then, eq. (5.1) may be applied to each entry. Finally, all resultant equations may be written in matrix form as

$$\left[\begin{array}{c} y^{(1)} \\ \vdots \\ y^{(j)} \\ \vdots \\ y^{(N)} \end{array} \right] = \left[\begin{array}{ccccc} g_1(\mathbf{x}^{(1)}) & \cdots & g_i(\mathbf{x}^{(1)}) & \cdots & g_M(\mathbf{x}^{(1)}) \\ \vdots & & \vdots & & \vdots \\ g_1(\mathbf{x}^{(j)}) & \cdots & g_i(\mathbf{x}^{(j)}) & \cdots & g_M(\mathbf{x}^{(j)}) \\ \vdots & & \vdots & & \vdots \\ g_1(\mathbf{x}^{(N)}) & \cdots & g_i(\mathbf{x}^{(N)}) & \cdots & g_M(\mathbf{x}^{(N)}) \end{array} \right] \left[\begin{array}{c} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_M \end{array} \right].$$

Please note that a superscript identifies a database entry and a subscript each different coefficient. In compact form this is

$$\mathrm{Y} = \mathbf{G}_X \mathbf{b},$$

where $\mathbf{G}_X$ depends on the basis functions and design $\mathbf{X}^{(N)}$. From this, the least-squares estimate for the coefficients is

$$\hat{\mathbf{b}} = \left(\mathbf{G}_X^\mathsf{T} \mathbf{G}_X\right)^{-1} \mathbf{G}_X^\mathsf{T} \mathbf{Y}.$$

Resulting from this regression, the covariance of parameters $\hat{\mathbf{b}}$ is

$$\mathrm{Cov}(\hat{\mathbf{b}}) = \sigma^2 \left(\mathbf{G}_X^\intercal \mathbf{G}_X\right)^{-1},$$

and the prediction variance is [60]

$$\mathrm{Var}(\hat{y}) = \sigma^2 \mathbf{g}(\mathbf{x})^\intercal \left(\mathbf{G}_X^\intercal \mathbf{G}_X\right)^{-1} \mathbf{g}(\mathbf{x}).$$

These assess the uncertainty on the parameters and estimate. As it becomes apparent, uncertainties depend on design $\mathbf{X}^{(N)}$ and the way it fills the basis functions space. Therefore, to come up with a design that considers uncertainty of the regression, the integrated variance could be used as index to assess a particular design. The integrated variance is [60]

$$I_{\mathrm{Var}} = \int_R \frac{1}{\sigma^2}\hat{y}\mathrm{d}\mu(\mathbf{x}) = \int_R \mathbf{g}(\mathbf{x})^\intercal \left(\mathbf{G}_X^\intercal \mathbf{G}_X\right)^{-1} \mathbf{g}(\mathbf{x})\mathrm{d}\mu(\mathbf{x}) = \mathrm{trace}\left(\mathbf{M}_R \left(\mathbf{G}_X^\intercal \mathbf{G}_X\right)^{-1}\right), \quad (5.2)$$

where

$$\mathbf{M}_R = \int_R \mathbf{g}(\mathbf{x})^\intercal \mathbf{g}(\mathbf{x})\mathrm{d}\mu(\mathbf{x}).$$

The index depends on the particular design $\mathbf{X}^{(N)}$ and the basis function's moment matrix $\mathbf{M}_R$ (refer back to section 4.3), that ultimately depends on which basis functions were used. Please note that here Lebesgue integration notation was used, as it is customary in statistics.

5.4 USEFUL DESIGNS

Different types of designs for exploring a spherical space ($\mathbb{S}^2$) are proposed here. In general, these designs could be regarded as zero-, one-, two-, and three-dimensional. Dimensionality here is understood as the way the points in the sphere are parametrised. Zero-dimensional designs come from selecting interesting, isolated points. One-dimensional designs are defined from line parametrisations, *e.g.*, using the sphere's canonical big circles through parametrisations such as (4.1). Two-dimensional dimensional designs come from surface parametrisations such as S_s^2 defined by (4.2) and S_c^2 defined by (4.3); in all cases they represent mappings from $\mathbb{R}^2$ to $\mathbb{S}^2$. Finally, three-dimensional designs come from distributing points directly in the sphere by using any distribution criterion, such as covering, packing, or any other.

5.4.1 ZERO-DIMENSIONAL DESIGNS

Zero-dimensional designs are *ad hoc* selections of important points. An immediate solution is to use the main axes' directions, *i.e.*, a three-point design with

$$\left\{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\intercal, \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\intercal, \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\intercal\right\},$$

and a six-point design where the set

$$\left\{ \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^{\mathsf{T}}, \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^{\mathsf{T}}, \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^{\mathsf{T}} \right\}$$

is added. This design is a *de facto* alternative for, *e.g.*, computing hydrodynamic loads or three-dimensional motion simulation. This is considered a reference design because any other design should be worth the effort of more computations, *i.e.*, should provide better-metric results.

5.4.2 ONE-DIMENSIONAL CIRCLE-BASED DESIGNS

One-dimensional designs are parametrised in terms of one parameter. These are designs that geometrically resemble points on a circle (line) in $\mathbb{S}^2$, so their parameter is an angle. More specifically, these could be designs where, *e.g.*, only xy-, xz-, and yz-plane points are considered. This type of designs are computed from circle parametrisations such as (4.1) of Sec. 4.1.1.

Designs based on circles could be computed by using an N_c-point uniform distribution on $\mathbb{S}^1$. This one is given by a uniform distribution of a parameter q such that $\{q_i\} = \left(\frac{i-1}{N_c}\right) 2\pi$, for $i = 1, \ldots, N_c$. From this an xy design is given by sets of points

$$\left\{ \begin{bmatrix} \cos q_i & \sin q_i & 0 \end{bmatrix}^{\mathsf{T}} \right\},$$

an xz design by

$$\left\{ \begin{bmatrix} \cos q_i & 0 & \sin q_i \end{bmatrix}^{\mathsf{T}} \right\},$$

and a yz design by

$$\left\{ \begin{bmatrix} 0 & \sin q_i & \cos q_i \end{bmatrix}^{\mathsf{T}} \right\}.$$

From these three circles one could explore a two-dimensional space by choosing one circle, *e.g.*, in the xy-plane: these will be called one-circle designs (1CD). One can define also a three-dimensional space by choosing, *e.g.*, the three circles: these will be called three-circle designs (3CD).

Furthermore, 3CDs (or any other design) could be combined and/or clipped to explore a whole, half, or eighth sphere. This is made to exploit any symmetry in the problem. To illustrate these designs, Fig. 5.4 (a) shows a complete 3CD, Fig. 5.4 (b) a half 3CD considering xz-plane symmetry, and Fig. 5.4 (b) an eighth 3CD considering all planes symmetry. Half-sphere 3CDs are important to this work because ROVs often have xz-plane symmetry.

Something important to take into account when considering such designs is the number of experimental points required. Table 5.3 addresses the amount of required experimental points

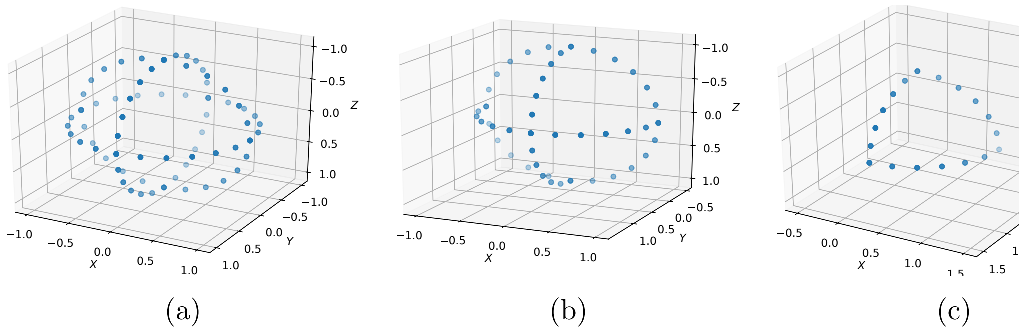


Figure 5.4. One-dimensional designs on a sphere

Table 5.3. Number of experimental points for three-circle designs

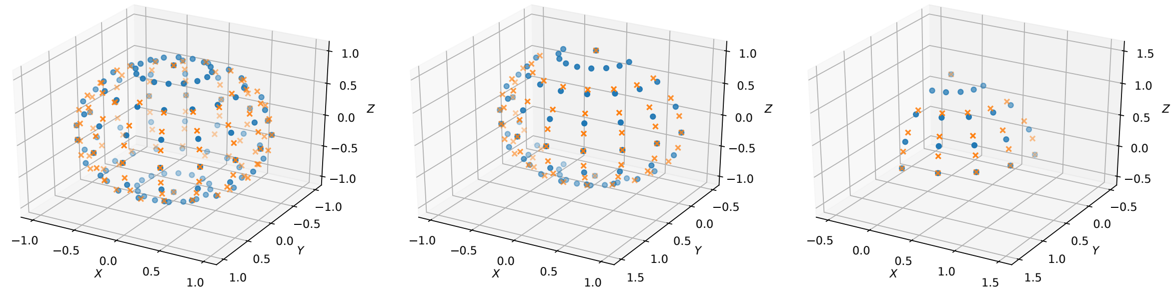
N_c	Δq_i	Eighth	Half	Whole
8	$\pi/4$	6	13	18
12	$\pi/6$	9	21	30
16	$\pi/8$	12	29	42
20	$\pi/10$	15	37	54
24	$\pi/12$	18	45	66
28	$\pi/14$	21	53	78
32	$\pi/16$	24	61	90
36	$\pi/18$	27	69	102

from a base uniform distribution of parameter q, when considering complete, half, or eighth spheres. The number of experimental points shown eliminates all possible redundancies. The data in the table could be generalised by considering the expressions further shown in Table 5.4.

5.4.3 TWO-DIMENSIONAL FULL-FACTORIAL DESIGNS

As expected, two-dimensional designs are parametrised in terms of two parameters. These are designs that fill the $\mathbb{S}^2$ space by using a two dimensional parametrisation such as S_s^2 defined by (4.2) and S_c^2 defined by (4.3). This parametrisation makes it possible to create a full factorial design (lattice or grid) in $\mathbb{R}^2$ and translate it to $\mathbb{S}^2$, similar to those shown in Figs. 4.1 and 4.2.

A representation of the experimental points for a whole, half, and eighth sphere is shown in Fig. 5.5. To compute the designs, n is chosen as the number of divisions on a quarter



(a) (b) (c)

Figure 5.5. Two-dimensional FFD on a sphere: (a) whole, (b) half, and (c) eighth. The • marker corresponds to S_s^2 and × marker to S_c^2

Table 5.4. Number of experimental points for each type of design

	Eighth	Half	Whole
3-circle	$3n$	$8n-3$	$12n-6$
Full factorial	n^2+n+1	$4n^2+1$	$8n^2-4n+2$

circle ($\pi/2$), *e.g.*, giving angle increments of $\pi/(2n)$. The plots in Fig. 5.5 illustrate as well a comparison of the distribution given by mappings S_s^2 and S_c^2. The two mappings give different distributions, with S_s^2 generating a greater concentration of points near the sphere's poles.

The number of experimental points could be computed from n, the amount of subdivisions made on a quarter circle ($\pi/2$). The expressions to compute these are shown in Table 5.4. Please note that to relate these designs to aforementioned circle designs the relation $n = N_c/4$ applies. These expressions allow one to illustrate how the number of experimental points increment when increasing the number of base subdivisions. An illustration of how the number of experimental points increases is shown in Fig. 5.6: one-dimensional designs grow linearly and two-dimensional quadratically. The plot is truncated at 100 experimental points: amounts around this value become impractical for time-consuming simulations. Furthermore, it becomes apparent how these full-factorial designs grow prohibitively large.

It is possible to use metrics to compare uniformity among three-circle (3CD) and full factorial (FFD) designs using spherical and cylindrical parametrisations over half sphere. Figure 5.7 shows maximin distance in terms of minimum separation angle (θ_{sep} in Sec. 4.1.3) and 2-energy (s-energy, with $s = 2$). Good distributions have big separation angles and low energy. From the plots, it becomes apparent that a FFD with cylindrical coordinates gives better distributions when increasing the number of experimental points. Consequently, from now on, S_c^2 (cylindrical coordinates) will be the preferred two-dimensional parametrisation. Thus,

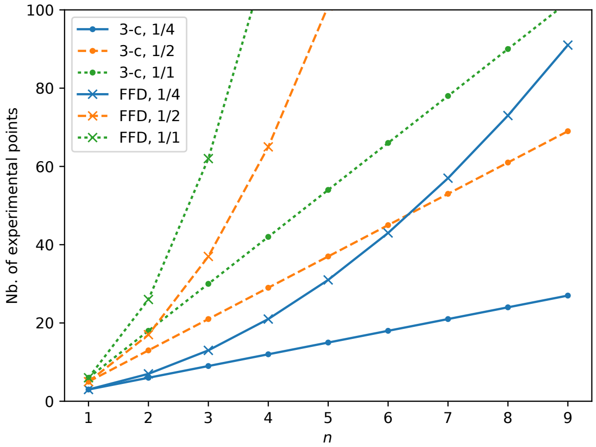


Figure 5.6. One-dimensional and two-dimensional designs comparison. Three circle designs are identified by *3-c* and spherical full-factorial designs by *FFD*

this parametrisation will serve as well as reference for comparing further two- and three-dimensional designs.

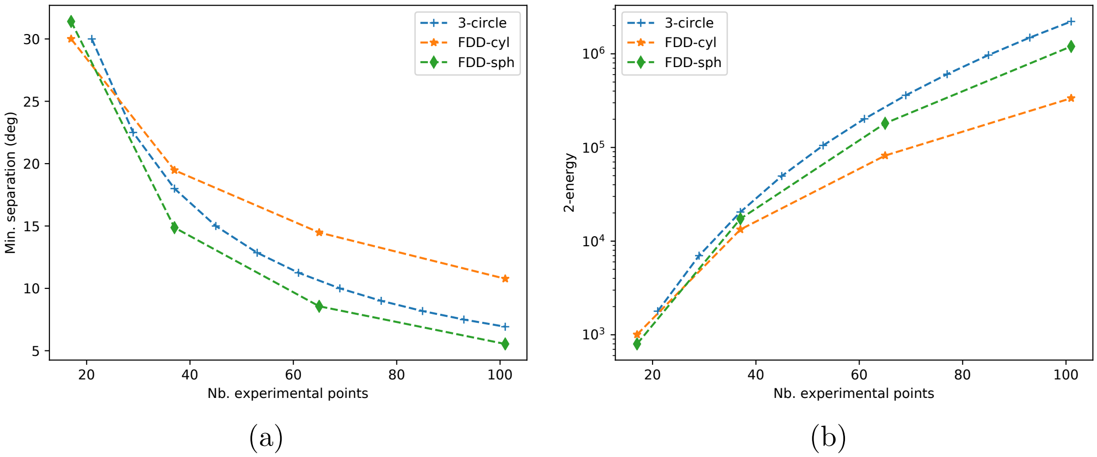


Figure 5.7. Metrics comparison for three-circle and spherical full-factorial designs over half sphere

5.4.4 TWO-DIMENSIONAL HYPERCUBE-BASED DESIGNS

The two-dimensional hypercube-based designs presented here are computed from an Orthogonal Hypercube Design (OHD), as shown in Sec. 5.2.3. The hypercube design table is mapped to a spherical domain by using mapping S_s^2 defined by (4.2) or S_c^2 defined by (4.3), but, as previously mentioned, the cylindrical parametrisation (S_c^2) is preferred. The designs are obtained from random permutations, but it is possible to select the best from a selected number

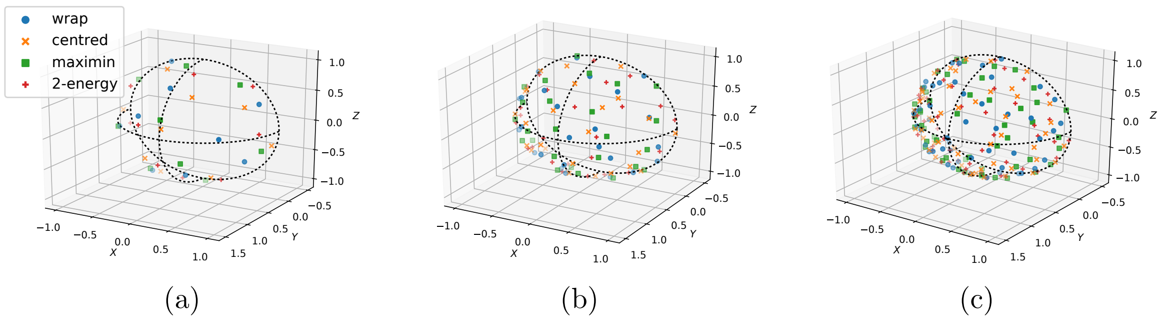


Figure 5.8. Two-dimensional OHD on a sphere when using different metrics to choose the best design: (a) 9-, (b) 25-, and (c) 49-point distributions.

of permutations according to different uniformity measures. The following considerations are taken into account:

- The best of 10^6 permutations is chosen.
- The domain is half sphere.
- Uniformity measures to select be best design are wrap-around discrepancy, centred discrepancy, minimum distance, and 2-energy. These uniformity measures are computed as shown in Sec. 5.2.4, considering the point distribution on a hypercube.

This gives way to OHD distributions that have different number of experimental points and exhibit different measures of uniformity. As an illustration, Fig. 5.8 shows distributions for 9, 25, and 49 experimental points, when choosing the best with respect each mentioned uniformity measure.

Furthermore, Fig. 5.9 shows the behaviour of each distribution when using uniformity measures over the sphere, as computed in Sec. 4.1.3, namely minimum distance and 2-energy. The measures of the OHDs are compared to a spherical FFD using a cylindrical parametrisation. From the plots it is possible to conclude that a FFD gives better uniformity metrics, meaning that OHD are more adequate for exploratory analyses when certain level of randomisation is desired. Conversely, OHD are not so adequate when a uniform exploration of the space is desired.

5.4.5 THREE-DIMENSIONAL SPACE-FILLING DESIGNS

Three-dimensional designs are obtained by distributing points directly in $\mathbb{S}^2$, by using criteria such as covering, packing, minimum-energy, or any other. These designs were discussed in

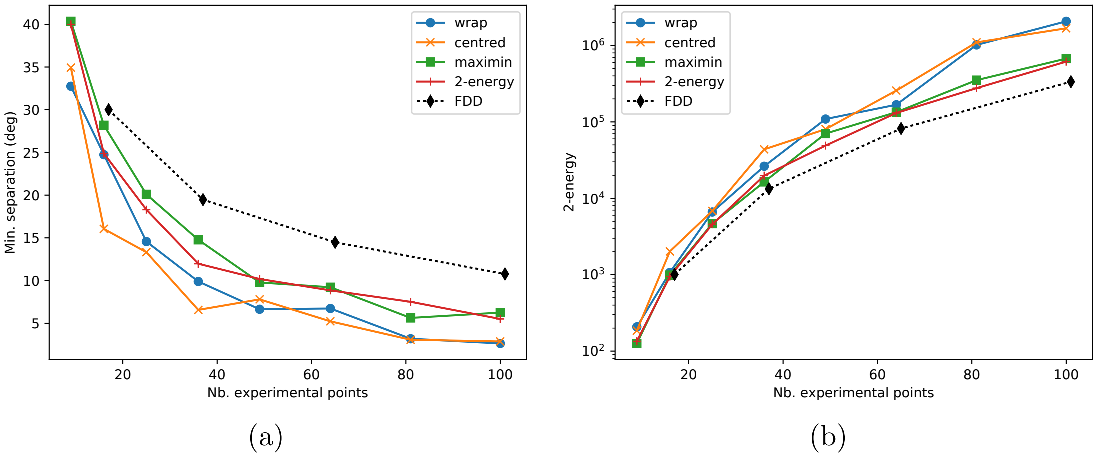


Figure 5.9. Metrics comparison for two-dimensional OHD over half sphere

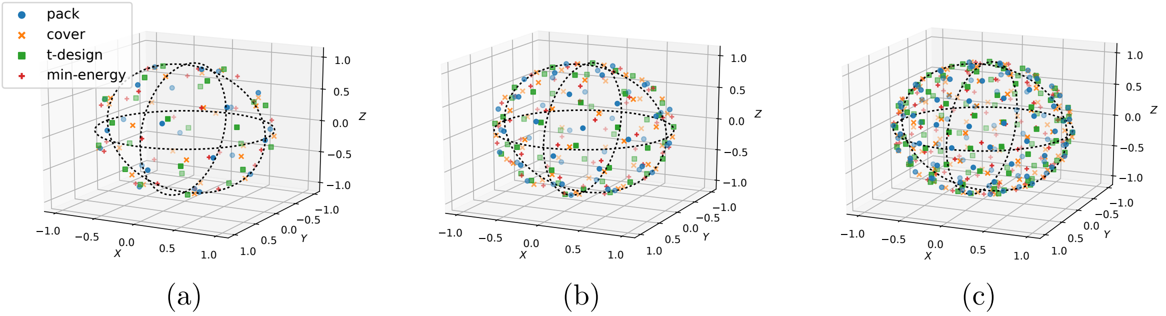
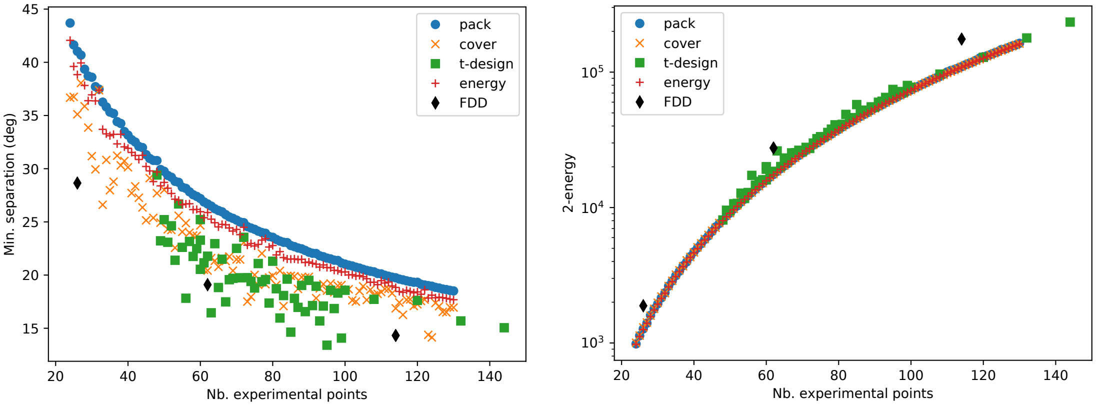


Figure 5.10. Three-dimensional designs on a whole sphere when using different types of distributions: (a) 24, (b) 48, and (c) 96 points.

Sec. 4.1.3 and are available [117, 115, 118, 116]. Four types of arrangements are considered, namely packings [117], coverings [115], t-designs [118], and minimal energy [116]. Some of these designs are illustrated in Fig. 5.10.

The uniformity of these designs is evaluated over arrangements on a whole sphere and using minimum separation angle (θ_{sep}) and 2-energy. These results are shown in Fig. 5.11. As one could expect, designs that optimise minimax distance deliver best (higher) separation angles; conversely, designs that optimise energy deliver lower energy. The graphs are compared to results obtained for FFD using cylindrical coordinates parametrisation: one could notice from Fig. 5.11 that three-dimensional designs give better metrics in all cases.



(a)   (b)

Figure 5.11. Metrics comparison for different three-dimensional designs over a whole sphere

5.4.6   DESIGN METRICS FOR META-MODEL REGRESSION

A simple, uncoupled model for drag forces may be written as

$$
\begin{aligned}
X &= X_{u|u|}u|u|,\\
Y &= Y_{v|v|}v|v|,\\
Z &= Z_{w|w|}w|w|.
\end{aligned}
$$

In this case the load meta-model bases are $B_{X_c} = \{u|u|\}$, $B_{Y_c} = \{v|v|\}$, and $B_{Z_c} = \{w|w|\}$. A metric based on integrated variance for N experimental points may be given from (5.2) as

$$
I_{\mathrm{Var}_u} = \frac{4\pi}{5}\left(\frac{1}{\sum_{i=1}^{N} u_i^4} + \frac{1}{\sum_{i=1}^{N} v_i^4} + \frac{1}{\sum_{i=1}^{N} w_i^4}\right).
$$

This metric sums the integrated variance related to all three loads meta-models.

A similar procedure may be followed for a model that takes into account cross-coupled effects, namely

$$
\begin{aligned}
X &= X_{u|u|}u|u| + X_{u|v|}u|v| + X_{u|w|}u|w|,\\
Y &= Y_{v|v|}v|v| + Y_{v|u|}v|u| + Y_{v|w|}v|w|,\\
Z &= Z_{w|w|}w|w| + Z_{w|u|}w|u| + Z_{w|v|}w|v|.
\end{aligned}
$$

In this case the loads meta-model bases are $B_{X_c} = \{u|u|,\, u|v|,\, u|w|\}$, $B_{Y_c} = \{v|v|,\, v|u|,\, v|w|\}$, $B_{Z_c} = \{w|w|,\, w|u|,\, w|v|\}$. The metric based on integrated variance for N experimental points, that sums the all three loads effects, may be given from (5.2) as

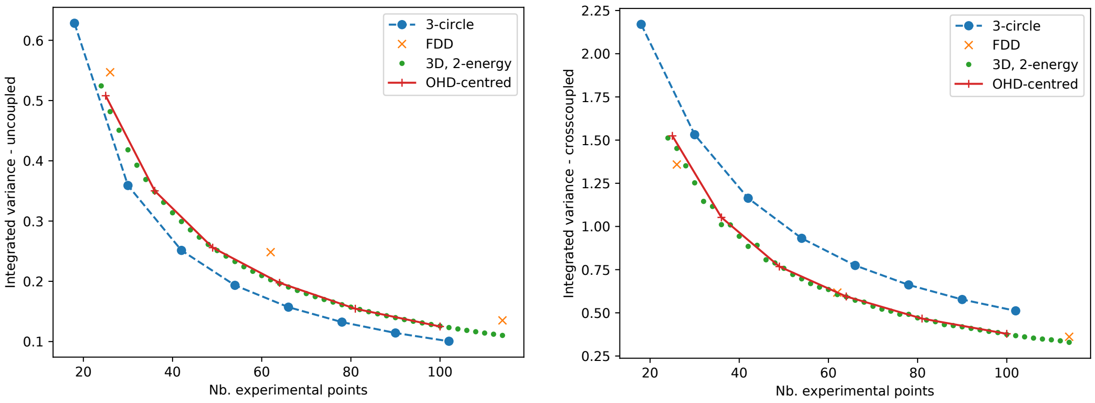
$$
I_{\mathrm{Var}_c} = \mathrm{trace}\left(\mathbf{M}_R\left(\left(\mathbf{G}_X^{\intercal}\mathbf{G}_X\right)^{-1} + \left(\mathbf{G}_Y^{\intercal}\mathbf{G}_Y\right)^{-1} + \left(\mathbf{G}_Z^{\intercal}\mathbf{G}_Z\right)^{-1}\right)\right),
$$

where

$$\mathbf{M}_R = \frac{1}{15}\begin{bmatrix} 12\pi & 16 & 16 \\ 16 & 4\pi & 8 \\ 16 & 8 & 4\pi \end{bmatrix}, \quad \mathbf{G}_X = \begin{bmatrix} u_1|u_1| & u_1|v_1| & u_1|w_1| \\ \vdots & \vdots & \vdots \\ u_N|u_N| & u_N|v_N| & u_N|w_N| \end{bmatrix},$$

$$\mathbf{G}_Y = \begin{bmatrix} v_1|v_1| & v_1|u_1| & v_1|w_1| \\ \vdots & \vdots & \vdots \\ v_N|v_N| & v_N|u_N| & v_N|w_N| \end{bmatrix}, \quad \text{and} \quad \mathbf{G}_Z = \begin{bmatrix} w_1|w_1| & w_1|u_1| & w_1|v_1| \\ \vdots & \vdots & \vdots \\ w_N|w_N| & w_N|u_N| & w_N|v_N| \end{bmatrix}.$$

Figure 5.12 shows a comparison of the aforementioned indices for selected designs over a whole sphere and different number of experimental points, namely 3CD, FDD using cylindrical coordinates, OHD selected from best centred discrepancy, and minimal-energy three-dimensional distribution. Worth mentioning, when the model is uncoupled, 3CDs show the best behaviour compared to other types of designs; conversely, when the model is cross-coupled, 3CD shows the worst behaviour. Moreover, for the cross-coupled model and distributions different to 3CD the integrated variance is improved mainly by increasing the number of experimental points: the difference among designs is small.



(a) (b)

Figure 5.12. Integrated variance comparison for different designs over a whole sphere: (a) uncoupled model and (b) cross-coupled model.

Part II

Hydrodynamics and cable modelling

Chapter 6

Hydrodynamics

This chapter develops an approach to study ROV hydrodynamics and manoeuvring by means of viscous-flow computations and time-domain simulations. In this approach, a Reynolds-Averaged Navier-Stokes (RANS) equations solver is used to apply the virtual captive test (VCT) approach, where experimental fluid dynamics (EFD) conditions are simulated to obtain the forces and moments exerted by the fluid onto the vehicle at certain flow conditions. Here, the work by Toxopeus [126], that applied this approach to ships and submarines (DARPA SUBOFF), is used as baseline and adapted to include ROVs' particularities. The data available from the computations are assembled into a meta-model that computes forces and moments faster. Then, the meta-model allows one to implement time-domain simulations for manoeuvring and feedback control studies.

The main objective consists in finding a simplified model for ROV's hydrodynamic loads, expressed through a regression function or meta-model obtained from a set of viscous flow computations. By using a modular approach [126], the idea is to compute forces and moments on the bare hull (ROV body), and then include other loads independently, such as thruster forces. The thruster forces may be obtained from models such as the Wageningen propeller series [74] and expressed through the model given by Eqs. (2.18) and (2.19). Given the many different phenomena possible, this work will focus on including drag considering three-dimensional relative motion, uncoupled rotational roll-pitch-yaw moments, and uncoupled added mass, which are the most relevant phenomena to include into an ROV model [48].

6.1 METHODOLOGY OVERVIEW

In order to study ROV hydrodynamics as proposed, a four-step methodology is suggested. As illustrated in Fig. 6.1, the four main stages are 1) computer experiments design, 2) viscous-flow computations, 3) meta-model assembly, and 4) time-domain manoeuvring simulations. In brief, the three first stages look forward to obtaining a proper hydrodynamics meta-model in a systematic manner; the fourth stage uses the obtained meta-model for further time-domain motion simulation.

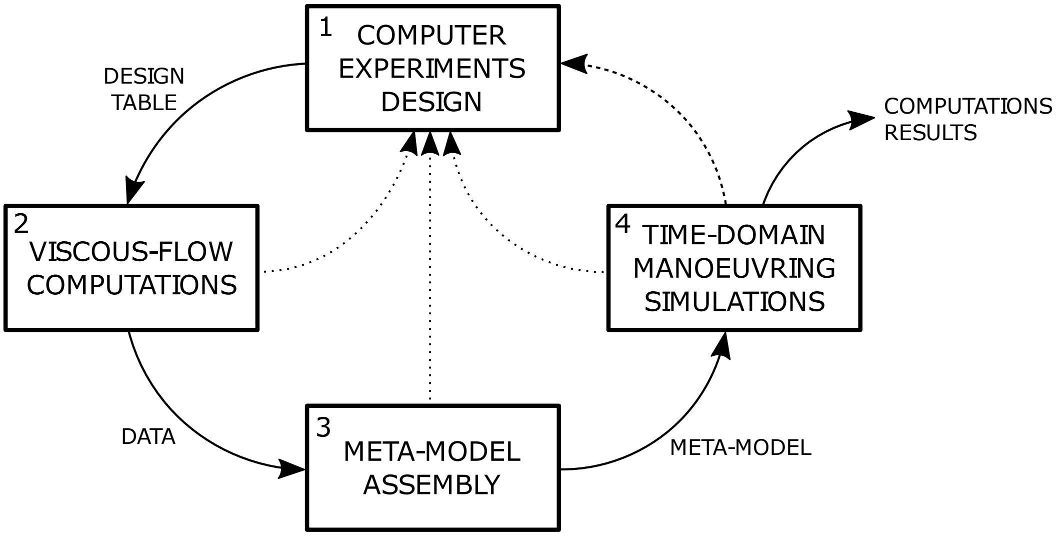

1 COMPUTER EXPERIMENTS DESIGN
DESIGN TABLE
2 VISCOUS-FLOW COMPUTATIONS
DATA
3 META-MODEL ASSEMBLY
META-MODEL
4 TIME-DOMAIN MANOEUVRING SIMULATIONS
COMPUTATIONS RESULTS

Figure 6.1. Manoeuvring study methodology

Stage one, *computer experiments design*, aims to determining what viscous-flow computations have to be done. Its results are condensed in *design tables* that specify a list of computation scenarios and their corresponding input values. In Fig. 6.1, the dotted lines suggest that experiment designs are fed with knowledge from further stages. Stage two, *viscous-flow computations*, carries through the previously specified computation scenarios. This stage deals with common CFD discipline: obtaining a proper computational discretised domain, computing using CFD code, and post-processing the obtained data. The main result of this stage is a data-base that relates kinematic conditions and hydrodynamic loads. Stage three, *meta-model assembly*, deals with assembling a proper meta-model from the data at hand. The result of this stage is a mathematical description of the relationship between kinematic conditions and hydrodynamic loads. Stage four, *time-domain manoeuvring simulations*, makes use of the obtained meta-model into time-domain simulations that test different ROV motion scenarios. Depending on whether the obtained results and overall process are satisfactory, the results may be considered final or reconsidered in another computer experiment design.

6.2 CONCEPTUAL TOOLS FOR VISCOUS-FLOW COMPUTER EXPERIMENTS

6.2.1 A NOTE ON COMPUTER EXPERIMENT DESIGN'S WORK-FLOW

Computer experiments design is a stage of the overall methodology that aims to determining what viscous-flow computations have to be done. The results of this stage are condensed in *design tables* that specify the list of computation scenarios with their corresponding input values. In general, the design of computer experiments stage may be approached by a three-part work-flow: 1) conceptual design, 2) selection of DoCE techniques, and 3) computation of design tables. The first part, *conceptual design*, requires one to clarify what the problem to solve is and translate it in terms of simulation scenarios: more specifically, the selection of the viscous-flow scenarios to compute. The second part, selection of *DoCE techniques*, acknowledges the myriad methods available to construct design tables. Each formal DoCE method follows after a specific purpose; therefore, this part aims to selecting appropriate techniques for the problem at hand. Finally, the third part, consists in applying the selected technique to compute the *design table*.

6.2.2 A NOTE ON HYDRODYNAMICS DIMENSIONLESS QUANTITIES

As part of defining and understanding hydrodynamics computer experiments, dimensionless quantities are used, namely Reynolds and Keulegan-Carpenter numbers. The Reynolds number is a dimensionless quantity that measures the ratio between inertial and viscous flow effects. The number is given by

$$\mathrm{Re} = \frac{\rho u L}{\mu} = \frac{uL}{\nu},$$

where ρ is flow density, u flow speed, L a characteristic length, μ absolute viscosity, and ν relative viscosity.

The Keulegan-Carpenter number is a dimensionless quantity that measures the relative importance between drag and inertia forces in oscillatory flow conditions. The number is given by

$$K_C = \frac{VT}{L},$$

where V is flow velocity amplitude, T period, and L characteristic length.

6.2.3 COMPUTATION SCENARIOS

The first part of conceptual design is to clarify the problem to solve. The main problem is to carry through viscous-flow computations to gather data of kinematic conditions and loads, in order to develop a global hydrodynamics model to study time-domain motion. This model was introduced in Sec. 2.2.3.3 and given by (2.13). In the model, hydrodynamic forces and moments are functions of relative velocity and acceleration, *i.e.*, ν_r and $\dot{\nu}_r$: this is written as $\tau_H\left(\nu_r, \dot{\nu}_r\right)$. The model is often expressed in component form, meaning that

$$\tau_H = \begin{bmatrix} X & Y & Z & K & M & N \end{bmatrix}^{\mathsf{T}}.$$

This implies that hydrodynamic loads are treated as individual scalar functions of relative velocities and accelerations. Explicitly and generally, this is

$$\begin{aligned}
X =& X(u_r, v_r, w_r, p_r, q_r, r_r, \dot{u}_r, \dot{v}_r, \dot{w}_r, \dot{p}_r, \dot{q}_r, \dot{r}_r),\\
Y =& Y(u_r, v_r, w_r, p_r, q_r, r_r, \dot{u}_r, \dot{v}_r, \dot{w}_r, \dot{p}_r, \dot{q}_r, \dot{r}_r),\\
Z =& Z(u_r, v_r, w_r, p_r, q_r, r_r, \dot{u}_r, \dot{v}_r, \dot{w}_r, \dot{p}_r, \dot{q}_r, \dot{r}_r),\\
K =& K(u_r, v_r, w_r, p_r, q_r, r_r, \dot{u}_r, \dot{v}_r, \dot{w}_r, \dot{p}_r, \dot{q}_r, \dot{r}_r),\\
M =& M(u_r, v_r, w_r, p_r, q_r, r_r, \dot{u}_r, \dot{v}_r, \dot{w}_r, \dot{p}_r, \dot{q}_r, \dot{r}_r),\\
N =& N(u_r, v_r, w_r, p_r, q_r, r_r, \dot{u}_r, \dot{v}_r, \dot{w}_r, \dot{p}_r, \dot{q}_r, \dot{r}_r).
\end{aligned}$$

Geometrically, the hydrodynamics model is a map from the space of relative velocities and accelerations to the space of forces and moments. Considering this, viscous-flow computation scenarios are means to explore the space of relative velocities and accelerations, *i.e.*, the domain set; consequently, the results of the computations are the corresponding image set elements. Then, the meta-model is an approximation of the map between the domain and image set elements that can be obtained from the data. All told, the selection of the type of viscous-flow computations is related to the way the domain set is explored.

Choosing the type of simulation scenarios relies on what can be computed using viscous-flow simulation code: CFD code capabilities and gathered experience. Among the computation options, one can find, *e.g.*, steady-state computation of forces and moments at different incoming current directions, unsteady imposed rotation computations, and imposed harmonic displacement computations. These types of viscous-flow scenarios are selected based on previous gathered experience (see *e.g.* [98]).

6.2.3.1 Steady-state flow computations at different inflow directions

Steady-state flow computations at different inflow directions are scenarios that assume a constant flow around the vehicle. When considering different directions of inflow velocity, these computations allow one to study loads as a function of velocities around the main axes as well as non-linear cross-coupling effects between these velocities. In this case, the relative velocity around the vehicle can be parametrised by

$$\begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = U\mathbf{e}_{\nu_1} = U \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix},$$

where U is the magnitude and e_u, e_v, and e_w are director cosines. As one could expect, the design tables related to computations selecting various e_u, e_v, and e_w are defined by using point distributions on $\mathbb{S}^2$, as proposed in Chapter 5 and further illustrated in Sec. 5.4.

Moreover, forces and moments can be made non-dimensional, *i.e.*, for a generic force F and moment M one has coefficients C_F and C_M such that

$$C_F = \frac{F}{\frac{1}{2}\rho A U^2} \quad \text{and} \quad C_M = \frac{M}{\frac{1}{2}\rho A L U^2},$$

where ρ is density, A is a projected area, and L is a length. This gives way to the definition of non-dimensional forces and moments

$$\begin{aligned} C_X &= \frac{X}{\frac{1}{2}\rho A_{yz} U^2}, & C_Y &= \frac{Y}{\frac{1}{2}\rho A_{xz} U^2}, & C_Z &= \frac{Z}{\frac{1}{2}\rho A_{xy} U^2}, \\ C_K &= \frac{K}{\frac{1}{2}\rho A_{yz} L_x U^2}, & C_M &= \frac{M}{\frac{1}{2}\rho A_{xz} L_y U^2}, & C_N &= \frac{N}{\frac{1}{2}\rho A_{xy} L_z U^2}; \end{aligned}$$

where A_{xy}, A_{xz}, and A_{yz} are projected areas and L_x, L_y, and L_z are lengths. Figure 6.2 shows how these areas and lengths are defined. This non-dimensional description assumes that the coefficients are defined for a determined flow regime. For an ROV, the flow regime usually ranges around Reynolds numbers in the order of 10^5–10^6.

This given, non-dimensional forces and moments become a function of the director cosines solely, such that one can write

$$C_{(\cdot)} = C_{(\cdot)}(e_u, e_v, e_w).$$

For instance, for x-axis non-dimensional force one can write a meta-model

$$C_X(e_u, e_v, e_w) = C_{X,u|u|} e_u|e_u| + C_{X,u|v|} e_u|e_v| + C_{X,u|w|} e_u|e_w| + \cdots, \tag{6.1}$$

such that the x-axis force can be computed as

$$X = \frac{1}{2}\rho A_{yz} U^2 C_X(e_u, e_v, e_w).$$

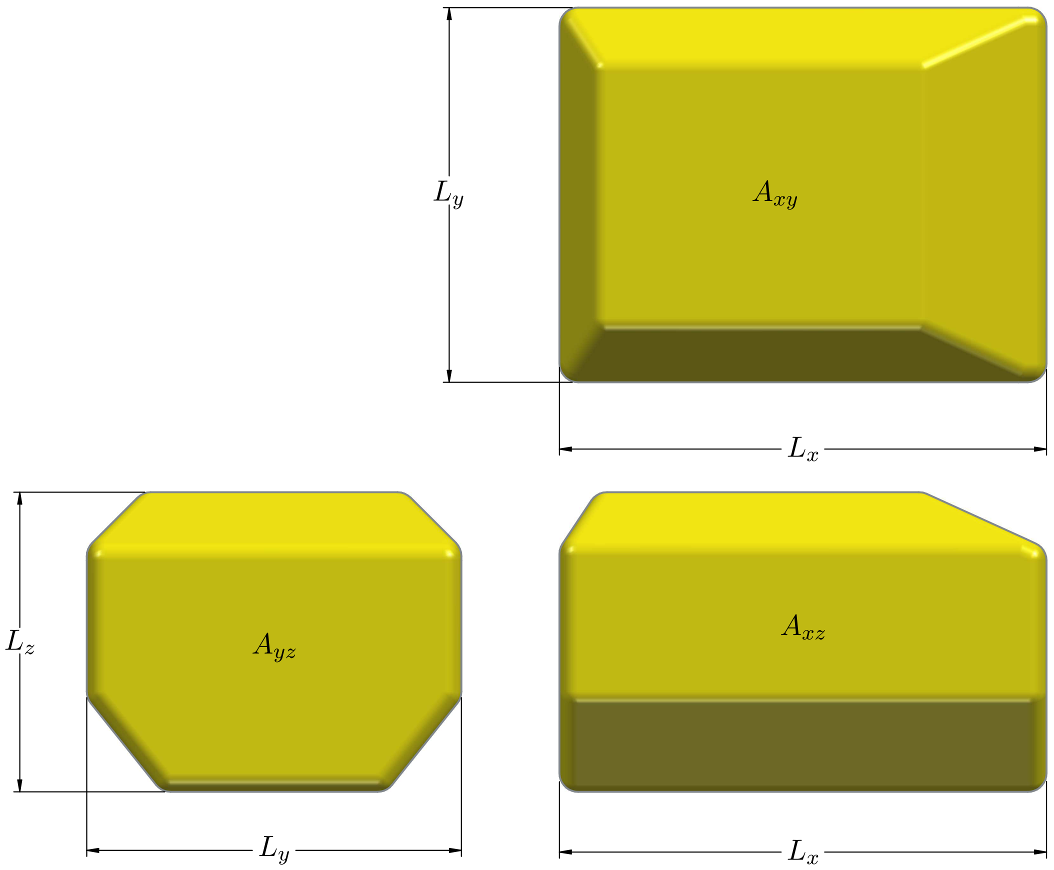


Figure 6.2. Areas and lengths for non-dimensional forces and moments

Please note that this meta-model only takes into account dependence on relative flow direction. Please note as well the use of the ellipsis: here and from this point on, this will mean that the model shown is only one part and not the whole model.

6.2.3.2 Steady-state flow computations for circular manoeuvres in the plane

Steady-state flow computations for circular manoeuvres in the plane are scenarios that assume a constant flow around the vehicle while rotating at a constant angular speed. In this case, the velocity vector lies in a plane perpendicular to the angular speed vector. This type of scenarios allow one to study cross-coupling effects between linear velocities and angular rates. Moreover, when the manoeuvre is performed in the horizontal plane, the relative velocity is given by $u = Ue_u$, $v = Ue_v$, and $w = 0$; additionally, the only angular velocity component is yaw r. In horizontal plane motion, these scenarios allow one to compute cross-coupling effects among surge, sway, and yaw velocities.

To make the analysis non-dimensional, the non-dimensional yaw rate is defined as $\gamma_r = rL_x/U$, so that yaw velocity is given by $r = \frac{U\gamma_r}{L_x}$. In other words, this parametrisation forces circular motion with speed U and radius $R = L_x/\gamma_r$. Moreover, director cosines may be parametrised by $e_u = \cos\beta$ and $e_v = \sin\beta$, where β is drift angle. With this information, these computations can be parametrised by a two-dimensional hypercube, *i.e.*, $\mathbb{C}_1^2$ as illustrated in Sec. 5.2.

Taking these ideas into account, non-dimensional forces and moments become a function of the director cosines e_u and e_v and non-dimensional rate γ_r. For instance, for x-axis non-dimensional force one can write a meta-model such as

$$C_X(e_u, e_v, \gamma_r) = \cdots + C_{X,u|r|}e_u|\gamma_r| + C_{X,vr}e_v\gamma_r + \cdots$$

and for y-axis non-dimensional force one can write a meta-model such as

$$C_Y(e_u, e_v, \gamma_r) = \cdots + C_{Y,v|r|}e_v|\gamma_r| + C_{Y,ur}e_u\gamma_r + \cdots .$$

Please note that this meta-model only takes into account cross-dependence between horizontal plane flow direction and yaw angular rate.

6.2.3.3 *Unsteady flow computations using harmonic (or arbitrary) imposed motion in translation and/or rotation*

When performing unsteady flow computations using harmonic imposed motion in translation or rotation, velocity or angular rate is variable. This means that these computations study loads dependence on velocity and acceleration. A common way to formulate such dependence is through Morison's equation [9]. For instance, x-axis force exerted by the fluid may be expressed as

$$X = \rho\nabla C_{X,\dot{u}}\dot{u} + \frac{1}{2}\rho A_{yz}C_{X,u|u|}u|u|$$

or, equivalently, using manoeuvring coefficients as

$$X = X_{\dot{u}}\dot{u} + X_{u|u|}u|u|.$$

Analogously, for x-axis (rolling) moment one has

$$K = \rho\nabla L_x^2 C_{K,\dot{p}}\dot{p} + \frac{1}{2}\rho A_{yz}L_x^3 C_{K,p|p|}p|p|$$

and using manoeuvring coefficients as

$$K = K_{\dot{p}}\dot{p} + N_{p|p|}p|p| + N_p p.$$

Computations to obtain these coefficients customarily impose harmonic motion, *e.g.*, for x-direction displacement one has

$$x(t) = x_{\max} \sin\left(\tfrac{2\pi}{T} t\right),$$

where $x_{\max}$ is motion amplitude and ω_x angular frequency; consequently, x-direction velocity is

$$u(t) = x_{\max} \tfrac{2\pi}{T} \cos\left(\tfrac{2\pi}{T} t\right),$$

where velocity amplitude is $u_{\max} = x_{\max} \frac{2\pi}{T}$. Equivalently for K, one has that rolling angle is

$$\phi(t) = \phi_{\max} \sin\left(\tfrac{2\pi}{T} t\right)$$

and rolling rate is

$$p(t) = \phi_{\max} \tfrac{2\pi}{T} \cos\left(\tfrac{2\pi}{T} t\right),$$

where the rolling rate amplitude is $p_{\max} = \theta_{\max} \frac{2\pi}{T}$. As expected, this type of computations is primarily parametrised in terms of the frequency and amplitude of the imposed motion, *e.g.*, $x_{\max}$ and $\phi_{\max}$ and T. Related to these parameters is Keulegan-Carpenter number, which for X is

$$K_{C_X} = \frac{u_{\max} T}{L_x} = 2\pi \frac{x_{\max}}{L_x}$$

and for K is

$$K_{C_K} = p_{\max} T = 2\pi \phi_{\max}.$$

It is important to acknowledge that, besides in-line motion around the main axes, the direction of the imposed motions could be parametrised as well by using point distributions on $\mathbb{S}^2$, as it was done in Sec. 6.2.3.1. Because unsteady computations are intrinsically expensive, it becomes difficult to pursue this road.

6.2.4 COMPUTATIONS UNCERTAINTY

Numerical computations on a mathematical model carry *uncertainty* and *errors*. *Uncertainty* refers to determining a region where the solution lies within certain degree of certainty: an interval of confidence. Determining an *error* requires that a 'true' solution is available; this requires a comparison between the expected value of experimental data and computations. Those errors and uncertainties may come from the model itself and from how correctly the model is resolved through numerical computations. The former case refers to how accurately the model represents reality and is called *validation*. The latter case refers to the mathematical exercise of quantifying how well the numerical computation resolves the model and is called *verification* [40]. Validation assesses the model and verification the computation.

Regarding verification, a clear distinction must be made between verifying the code, namely *code verification*, or a particular solution, namely *solution verification*: code verification should precede solution verification. Code verification concentrates on assessing errors between a model and the code that solves the model. This procedure may be understood as a responsibility of those who write the code and is often approached practically by the method of the manufactured solutions (MMS) [39]. Conversely, solution verification is a responsibility of those who use the code to solve a particular problem. Solution verification require assessing errors and uncertainties during a particular computation.

Computation errors have three main sources [126]: round-off, iterative, and discretisation error. Round-off errors are commonly considered negligible because processors use double-precision arithmetic and the other errors are of greater significance. Iterative errors arise when the modelling equations are not resolved enough along iterations [40]. Spatial discretisation is usually quantified by performing grid refinement studies [38]. Because validation measures how well models represent reality, it requires the comparison of model predictions with experimental data coming from physical models. Validation requires measuring numerical, experimental, and parameter uncertainties and errors. Theory and examples on solution verification and validation are available in various sources, see for instance [40, 126, 95].

Generally, uncertainty is approached as a point-wise problem, *i.e.*, assessed for each unique computation scenario [126, 38]. As mentioned, the three main sources of error in computations are round-off, iterative, and discretisation errors. Round-off error are due to arithmetic machine precision: they become important for highly refined grids and often are neglected. Iterative errors come from lack of equation resolution along iterations; this is because the model equations are non-linear. Iterative errors are assessed at each individual computation by checking the results at each iteration step. Finally, discretisation errors are due to the approximation of continuum differential equations into approximated discrete-algebraic equations. These last errors are of special interest because, unlike the past two, they are reduced by improving the computational domain.

The discretisation error ϵ_ϕ is assessed from a variable ϕ that stands for any integral or other functional of a local flow quantity. In this work, ϕ represents forces and moments integrated over the body surface for any given computation. A practical solution to compute the discretisation error is by assuming that the error follows some predictable, convergent behaviour: the computations approach the exact solution by improving the grid. The error behaviour is often predicted by using Richardson extrapolation, *i.e.*,

$$\epsilon_\phi \approx \delta_{\mathrm{RE}} = \phi - \phi_0 = \alpha h^p, \tag{6.2}$$

where ϕ_0 is an estimation of the exact solution, h the typical cell size, α a constant, and p

the order of convergence. If p is known or assumed two grids suffice to find α and ϕ_0, when p is unknown three grids suffice. Nevertheless, as shown in [38], in practical applications it is better to find the parameters from four (or more) grids using least-squares estimation. Because the discretisation error is unknown, the discretisation uncertainty is computed as the absolute value of the error times a factor of safety, *i.e.*,

$$U_\phi = |\epsilon_\phi| F_s.$$

Often, $F_s = 1.25$ when p is close to 2 and $F_s = 3$ otherwise. This is because p should reflect the fact that numerical methods often have order-2 accuracy [38].

Whenever ϕ does not show asymptotic behaviour, uncertainty is quantified by using Student's t-distribution [126]. By using this distribution, a confidence interval $[\phi_{\text{min}}, \phi_{\text{max}}]$ is obtained by using the 95 % percentile. From this, uncertainty is computed as

$$U_\phi = \tfrac{1}{2}\left(\phi_{\text{max}} - \phi_{\text{min}}\right).$$

Iterative uncertainty is denoted as U_i and computed as

$$U_i = \tfrac{1}{2}\left(\max(\phi) - \min(\phi)\right).$$

This is done when ϕ shows oscillatory convergence along iterations. Given that ϕ is distributed inside an interval, the central value is determined from the median. In the computations to come, uncertainty and central value are computed from a number of n last iterations where ϕ shows apparent, sustained oscillations.

6.3 VISCOUS-FLOW COMPUTATIONS ON VISOR3

This section applies viscous-flow computations for obtaining a hydrodynamic forces and moments model for visor3 (refer back to Sec. 3.1 for a description of visor3). The viscous-flow solver used in this section is MARIN's ReFRESCO (that will be described later). The computations were carried out during an internship in MARIN during Fall 2015. In order to compute non-dimensional forces and moments, relevant lengths and projected areas of visor3 are specified in Table 6.1 and illustrated in Fig. 6.3.

6.3.1 REFRESCO

ReFRESCO is a viscous-flow CFD code that solves multiphase (unsteady) incompressible flows using the Navier-Stokes equations, complemented with turbulence models, cavitation

Table 6.1. VISOR3 areas and lengths values for non-dimensional forces and moments computations

Parameter	Value	Units
A_{xy}	0.27843306	m^2
A_{yz}	0.18735244	m^2
A_{xz}	0.25253246	m^2
L_x	0.66	m
L_y	0.70	m
L_z	0.69	m

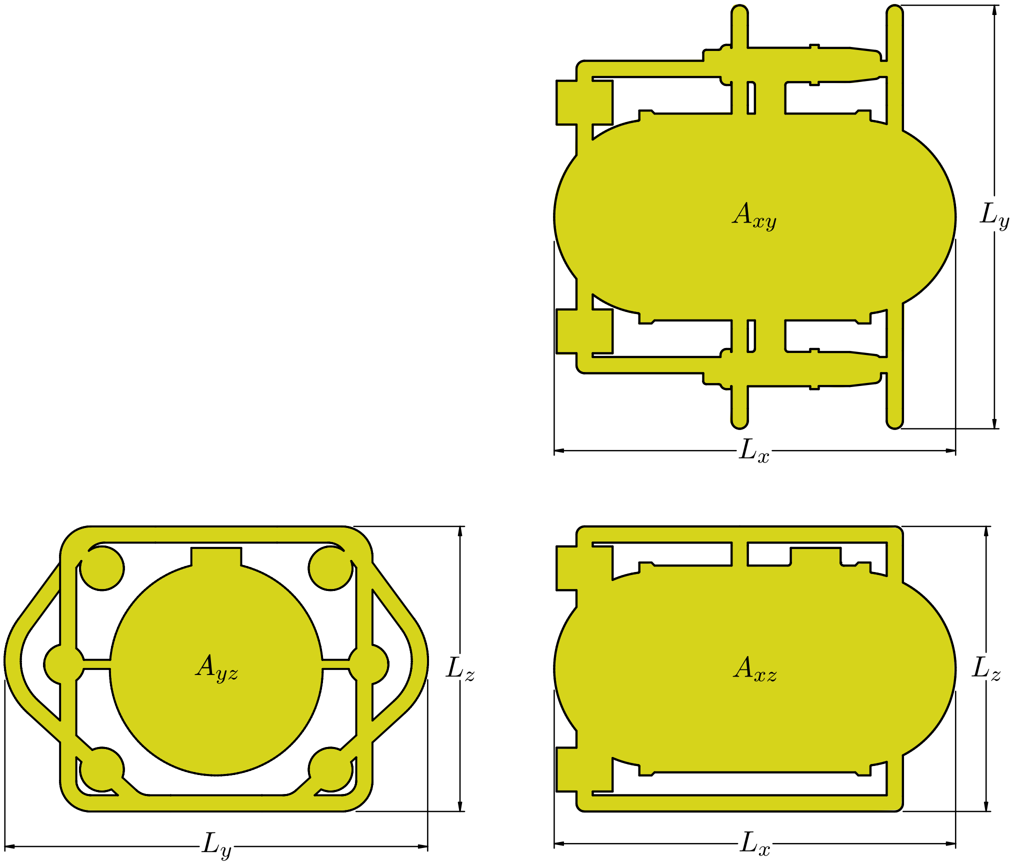


Figure 6.3. VISOR3 areas and lengths description for non-dimensional forces and moments computations

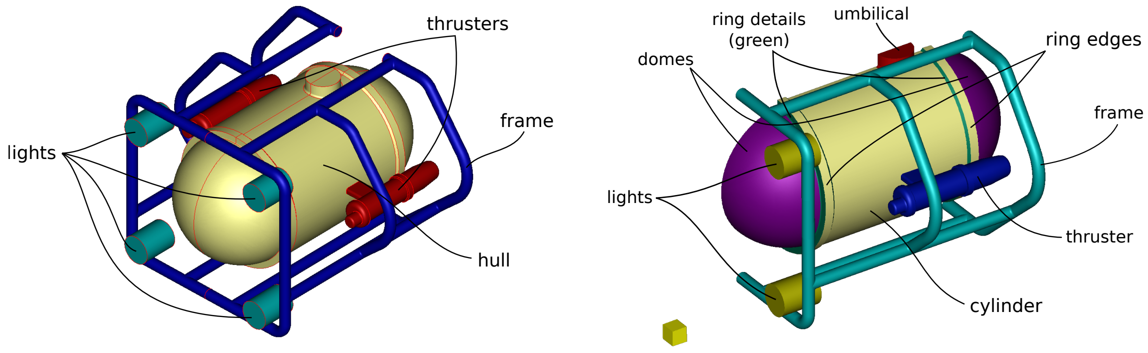
models, and volume-fraction transport equations for different phases [134]. The equations are discretised using a finite-volume approach with cell-centred collocated variables, in strong-conservation form, and a pressure-correction equation based on the SIMPLE algorithm is used to ensure mass conservation [71]. Time integration is implicit using first or second-order backward schemes. At each implicit time step, the non-linear system for velocity and pressure is linearised with Picard's method and either a segregated or coupled approach is used. In the latter, the coupled linear system is solved with a matrix-free Krylov subspace method using a SIMPLE-type preconditioner [71]. A segregated approach is always adopted for the solution of all other transport equations. The implementation is face-based, which permits grids with elements consisting of an arbitrary number of faces (hexahedrals, tetrahedrals, prisms, pyramids, etc.), and if needed h-refinement (hanging nodes). State-of-the-art CFD features such as moving, sliding, and deforming grids, as well automatic grid refinement are also available. For turbulence modelling, RANS/URANS, SAS, and DES approaches can be used (PANS and LES are being currently studied). The code is parallelised using MPI and subdomain decomposition and runs on Linux workstations and HPC clusters. ReFRESCO is currently being developed, verified, and validated at MARIN (in the Netherlands) [37, 128, 73, 104, 141, 69] in collaboration with IST (in Portugal) [94], USP-TPN (University of Sao Paulo, Brasil) [105], TUDelft (Technical University of Delft, the Netherlands) [71], UoS (University of Southampton, UK) [61] and recently UTwente (University of Twente, the Netherlands) and Chalmers (Chalmers University, Sweden).

In these computations, the steady-state and unsteady Navier-Stokes (NS) equations were solved for single-phase turbulent incompressible flow using ReFRESCO at different scenarios. As a starting point, turbulence is modelled using Menter's Shear-Stress Transport model, as used in [127]. For the sake of clarity, steady-state computations will refer to using numerical solvers for the steady-state NS equations; conversely, unsteady computations will refer to using numerical solvers for the unsteady NS equations.

6.3.2 COMPUTATIONAL DOMAIN

In this work, Hexpress is used to obtain the computational domain. Hexpress is a non-structured, hexahedral mesh generator software for complex arbitrary geometries. The obtained meshes are non-conformal and body-fitted and are obtained in a volume-to-surface approach, where there is no need for the creation of a surface mesh. This software, for instance, allows one to create computational domains to be used in RANS computations from CAD imported files; it includes tools for mesh refinement and optimisation, as well as for the insertion of viscous layers. See for example [41], for an application of Hexpress, where the computational domain for analysing the hull of a ship is created.

visor3's computational domain is shown herein. The computational domain is a non-structured grid made of hexagonal elements that represents the body of water surrounding the ROV. Two geometries are of importance for defining the computational domain: the ROV's and outer boundary's. The geometry of the ROV's surface is shown in Fig. 6.4 (a). Here, this surface is always considered a no-slip boundary. The surface is a simplified geometry of the ROV that includes four lights, protecting frame, hull, and external thrusters. The hull is represented by a cylinder with two hemispheres, including also the most relevant section changes and a simplified tether connection. The external thrusters do not include nozzles and propellers; also the tunnel thrusters and their tunnels are not included. The lights are cylinders without section changes. Other geometrical features were considered negligible. Figure 6.4 (b) shows different regions on the surface. These regions are chosen so as to define differentiated grid refinement parameters: smaller geometrical features require more grid refining and vice versa.



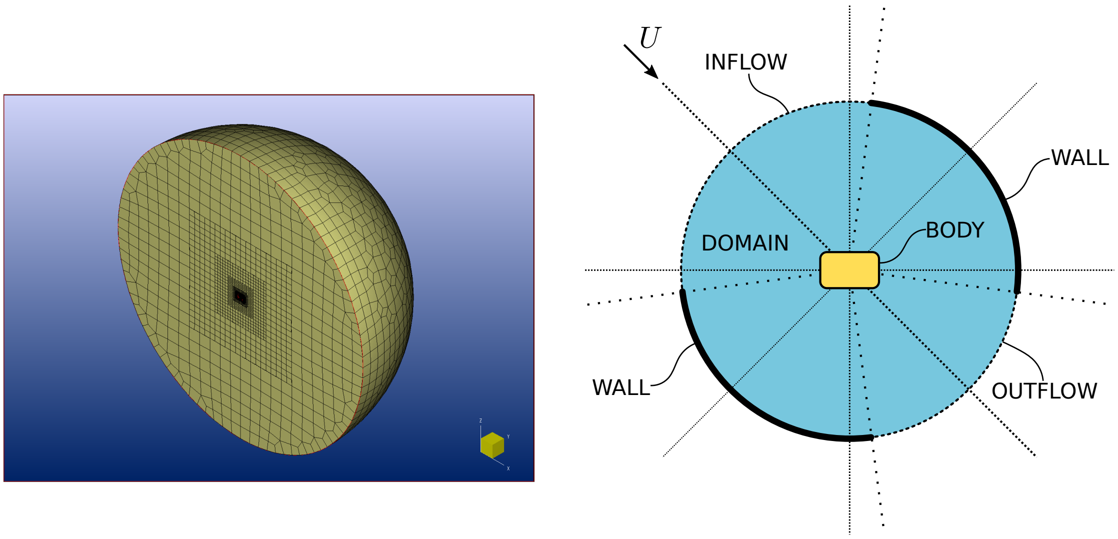
(a) (b)

Figure 6.4. VISOR3 simplified geometry and boundary conditions

A spherical geometry was used for the outer boundary: this geometry allows different types of computations keeping the same grid. Two options for defining boundary conditions on the sphere were used. The first option is simpler from a CAD perspective: create a whole spherical surface and then the viscous flow solver defines what part of the surface is inflow, outflow, or wall. Figure 6.5 (a) shows how half a domain may look like and Fig. 6.5 (b) shows a diagram of how the solver defines the outer boundaries depending on the inflow's relative direction: this is known as boundary auto-detection.

The second option consists in creating a divided sphere surface using CAD software. This divided sphere has bottom and top caps that are defined as wall boundaries. It also has lateral panels that are defined either as wall, inflow, or outflow boundaries depending on each computation's inflow direction. The sphere divisions are shown in Fig. 6.6.

Mesh preparation in Hexpress follows a series of steps: initial mesh, adaptation, snapping, optimisation, and addition of viscous layers. The *initial mesh* is a box-shaped grid, adaptation refines the grid around the boundaries, snapping trims the initial grid according to the



(a) (b)

Figure 6.5. Boundaries autodetection

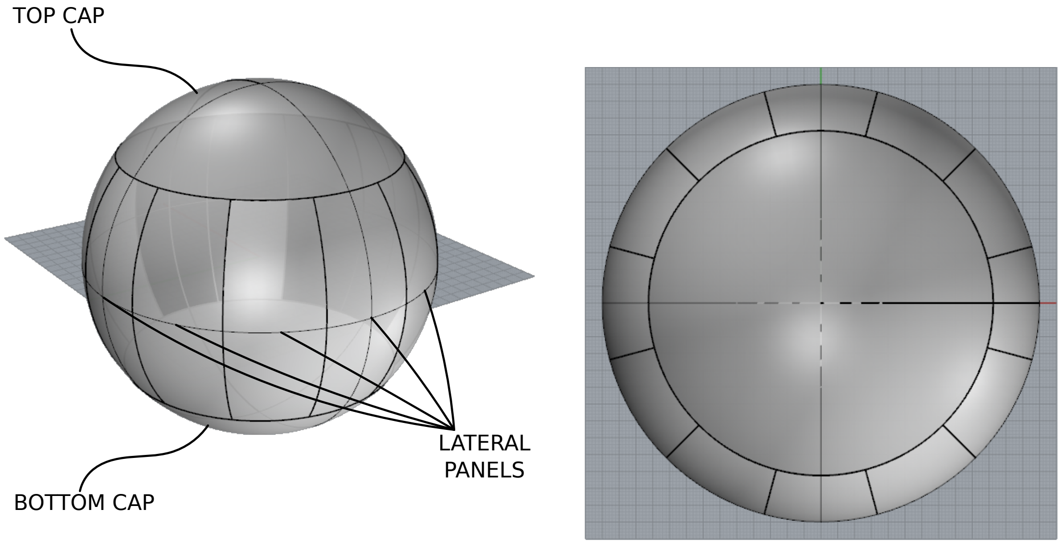
domain's boundaries and optimisations corrects the cells geometry, and the addition of viscous layers further refines the grid so the discretisation captures viscous flow near no-slip boundaries. VISOR3's mesh and, furthermore, any other ROV could be obtained from processing half the geometry, taking advantage from xz-plane symmetry.

The overall cell size and, thus, the number of elements are mainly affected by the initial mesh and adaptation settings. Figure 6.2 shows different values used for obtaining different VISOR3's grids (half grid).

Table 6.2. Initial mesh settings

Nb. cells total	1372	2048	4000	6912	16384	23328
Nb. cells x	14	16	20	24	32	36
Nb. cells y	7	8	10	12	16	18
Nb. cells z	14	16	20	24	32	36

To perform adaptation, Fig. 6.4 (b) shows the surfaces used and Table 6.3 HEXPRESS's parameters. For the sake of generalisation, N_r will be considered the maximum number of refinement steps. In this example, among all grids, this value ranges from 8 to 10, depending on the initial mesh size. Then, less refined cells will have either $N_r - 1$ or $N_r - 3$ refinement steps. This means that if the most refined surface has 10 refinements, other surfaces have



Perspective view

Top view

Figure 6.6. Division of the spherical outer domain

either 7 or 9 refinements.

Table 6.3. Adaptation refinement parameters

Surface group	Max. nb. of refinements	Adaptation criteria	Target cell size	Max. aspect ratio	Refinement diffusion
Domes	$N_r - 3$	Distance	0 0 0	2	3
Cylinder	$N_r - 3$	Distance	0 0 0	2	3
Ring edges	$N_r - 3$	Distance	0 0 0	2	3
Umbilical	$N_r - 3$	Distance	0 0 0	2	3
Lights	$N_r - 1$	Distance	0 0 0	2	3
Frame	$N_r - 1$	Distance	0 0 0	2	3
Thrusters	N_r	Distance	0 0 0	2	4
Ring details	N_r	Distance	0 0 0	2	3

To perform snapping and optimisation, few settings diverged from the default in these two steps. There were no changes in the default during snapping and there were two main changes in the optimisation settings: first, the number of optimisation steps were chosen as four and, second, the minimum orthogonality angle threshold was chosen as 35 deg. This often prevented final cell angles below 20 deg. The complete parameters used are shown in Table 6.4.

To add viscous layers, Hexpress uses a simplified procedure for the user to define the amount

Table 6.4. Grid optimisation parameters

Parameter	Value
Max. nb. of external optimisation	4
Max. nb. of invalid cells	100
Max. nb. of final optimisation iterations	7
Percentage of vertices to optimise during final optimisation	3.0
Max. nb. of orthogonality optimisation iterations	4
Minimal optimisation threshold (deg)	35

of viscous layers required for each surface based on the Reynolds number and the expected y^+ value. These parameters are specified in Fig. 6.5. The expected y^+ value was varied among different grids, but in general the most common value, along with the other parameters, is shown in Table 6.5. Nevertheless, different computations were performed to determine a grid that has adequate y^+ values (see Sec. 6.3.3).

Table 6.5. Viscous layers parameters

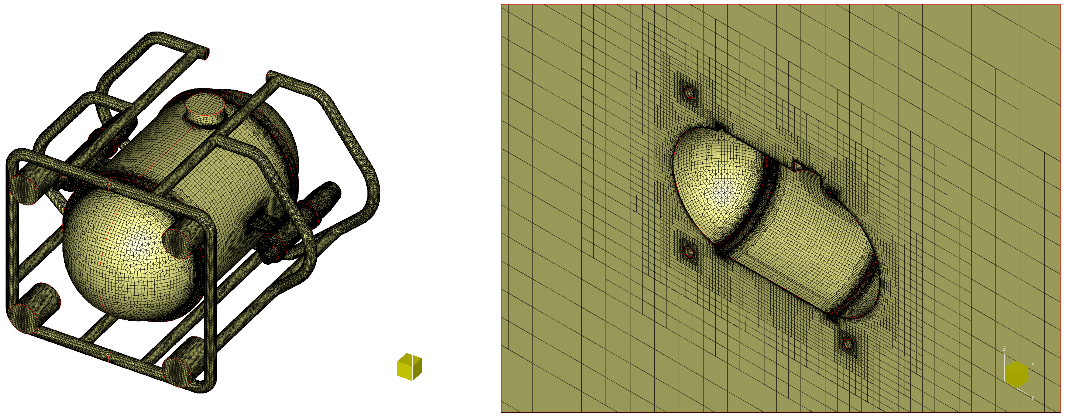
Parameter	Value	Units
Reference velocity	1.5	m/s
Reference length	0.66	m
Kinematic viscosity	1.002e-6	m^2/s
y^+	0.5	-

As expected, different settings will lead to different grids, even when there are just slight variations. Nevertheless, Fig. 6.7 shows how the grid looks like after all steps.

6.3.3 VERIFICATION COMPUTATIONS

To define an appropriate computational domain, different verification computations were carried out to study the influence of grid-related parameters such as domain size, mesh refinement, viscous layers, and flow regime. This is made to assess the uncertainty due to making computation-related decisions. These verification computations had the following goals:

- Definition of viscous layers so that y^+ is adequate.
- Definition of the size of the whole computational domain. Because the computation domain has a spheric outer boundary, this refers to choosing and adequate domain radius.
- Evaluation of different levels of grid refinement. In this case the main idea is to assess



(a) (b)

Figure 6.7. Resulting grid for VISOR3: (a) projection on the body surface and (b) central plane

the level of uncertainty related to using different grids with different number of cells.

- Evaluation of the effect of inflow velocity's magnitude in non-dimensional force. In other words, the idea is to verify how non-dimensional force varies with flow regime.

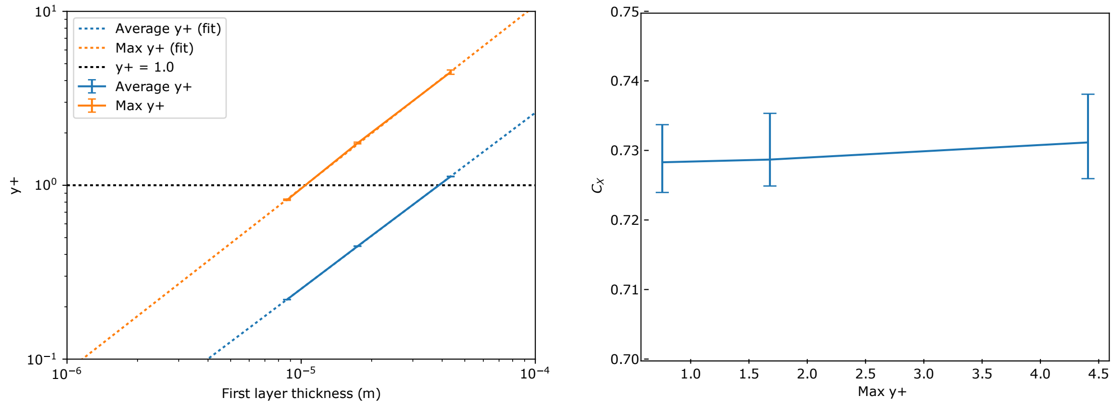
In all simulations a unique flow condition was imposed: constant inflow velocity along surge direction at 1.5 m/s. A slight exception is done when the effect of different magnitudes of inflow velocity is evaluated; in this part only flow direction is kept constant.

6.3.3.1 Viscous layers

When defining the computational domain, setting of the viscous layers is based on an initial guess given from reference velocity, reference length, kinematic viscosity, and expected y^+: Table 6.5 showed the typical values used. Here, three simulations were carried out using three grids with different values for the first layer's thickness in order to determine an adequate value for it. Table 6.6 shows some relevant parameters for each simulation. The results for maximum and average y^+ for the three options are shown in Fig. 6.8 (a). This plot shows dependence of y^+ on first layer thickness in log-log scale. Consequently, this result is used as guideline for choosing first layer's thickness in further computations. Moreover, Fig. 6.8 (b)

Table 6.6. Viscous layer analysis simulation parameters

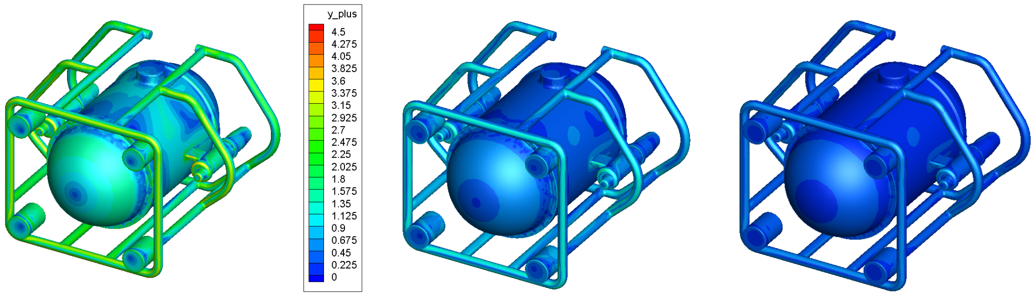
Case	1	2	3
First layer thickness (m)	4.35e-5	1.74e-5	8.70e-6
Grid's nb. of cells	10526132	12698946	14843124



(a) (b)

Figure 6.8. y^+ (a) at different first layer thicknesses and (b) relation between y^+ and non-dimensional force

illustrates how non-dimensional force in x-direction is affected when y^+ has values above and below 1. Because the main result from the simulations is body forces and moments, this figure indicates how uncertainty is introduced by non adequate y^+ values (*i.e.*, $y^+ > 1$). Furthermore, Fig. 6.9 illustrates how y^+ varies among all grids and shows where the most critical regions are.



Case 1 (less refined) Case 2 Case 3 (more refined)

Figure 6.9. y^+ visualisation in different grids

6.3.3.2 *Computational domain size*

Given that the external boundary of the domain is spherical, its size is measured using radius R and non-dimensional radius R/L_x. The radii used ranged from $5L_x$ to $22.7L_x$. The parameters used in each simulation case are show in Table 6.7. These settings were chosen so after N adaptations (and before snapping, optimising, and viscous layers addition) the cell size was almost the same in all cases.

Table 6.7. Computational domain size cases

Case	1	2	3	4	5	6	7
Radius (R/L_x)	22.7	15.2	15.2	10.6	9.5	7.6	5.0
Radius (m)	15	10	10	7	6.25	5	3.3
Init. grid nb. cells (half)	6912	2048	16384	5324	4000	2048	4000
Nb cells x	24	16	32	22	20	16	20
Nb cells y	12	8	16	11	10	8	10
Nb cells z	24	16	32	22	20	16	20
Max nb. of ref. N_r	10	10	9	9	9	9	8
Cell size after N_r ref. (m)	1.22e-3	1.22e-3	1.22e-3	1.24e-3	1.22e-3	1.22e-3	1.29e-3
Nb cells final (whole)	14.4M	14.4M	14.4M	13.8M	14.4M	14.4M	12.9M

Figure 6.10 shows values for non-dimensional force in surge direction at different domain radii, as well as a quantification of non-dimensional force uncertainty. Uncertainty shown in the figure is computed from iterative uncertainty U_i and U_{G_1} assuming asymptotic convergence. Total uncertainty is computed from Euclidean norm, assuming uncertainties are uncorrelated, as

$$U = \sqrt{U_i^2 + U_{G_1}^2}.$$

The plots show how uncertainty behaves when using smaller domains. This indicates that for the smallest domain, uncertainty is dominated by domain size; conversely, from that size on, iterative uncertainty is dominant. Based on these results, in further simulations, a radius of $9.5L_x$ is used.

6.3.3.3 Mesh refinement

Computations to estimate uncertainty due to grid refinement were carried out. Table 6.8 shows some of the parameters used for creating grids with different levels of refinement. Figure 6.11 shows results for non-dimensional force in x-direction, as well as an estimation of uncertainty. Uncertainty shown in the figure is computed from iterative uncertainty U_i, uncertainty due to domain size U_{G_1}, and uncertainty due to level of refinement U_{G_2}. Domain size uncertainty was obtained in Sec. 6.3.3.2. Uncertainty due to level of refinement is obtained by computing 95 % percentile uncertainty from Student's t-distribution of the medians. As before, total uncertainty is computed from Euclidean norm, assuming uncertainties are uncorrelated, as

$$U = \sqrt{U_i^2 + U_{G_1}^2 + U_{G_2}^2}.$$

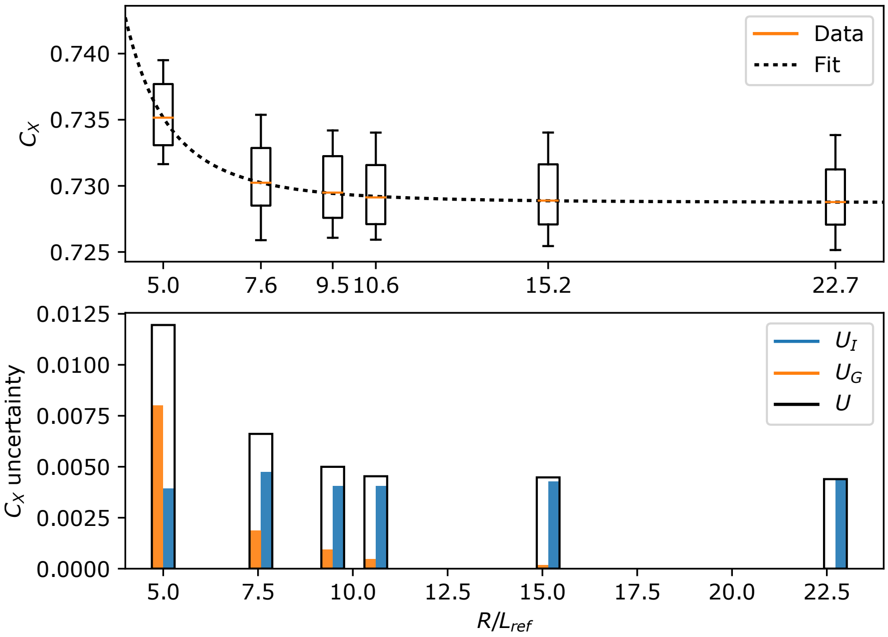


Figure 6.10. Behaviour of non-dimensional force and uncertainty at different domain sizes

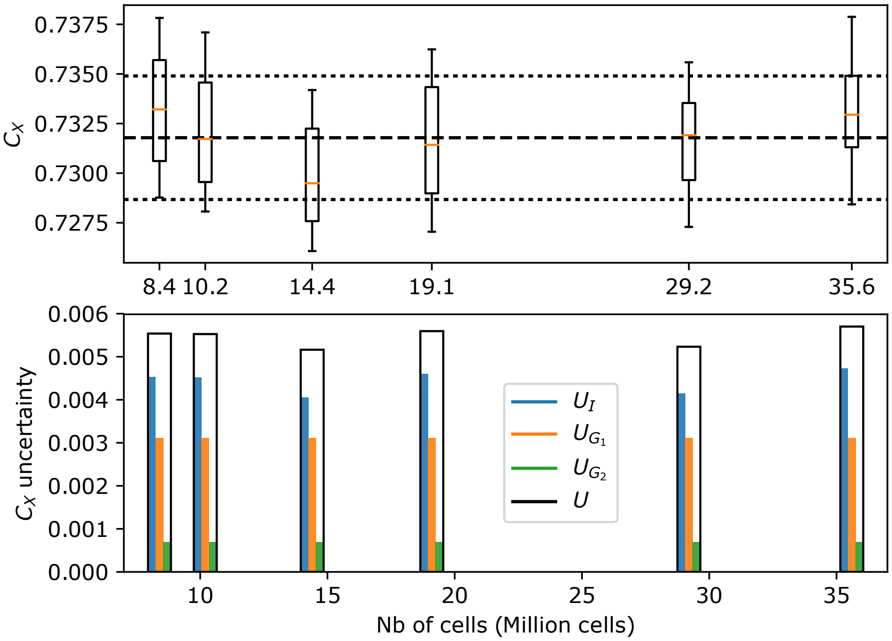


Figure 6.11. Non-dimensional force at different levels of refinement

Table 6.8. Mesh refinement settings

Case	1	2	3	4	5	6
Domain radius (m)	6.25	6.25	6.25	6.25	6.25	6.25
Nb cells init mesh (half)	1372	2048	4000	6912	16384	23328
Nb cells x	14	16	20	24	32	36
Nb cells y	7	8	10	12	16	18
Nb cells z	14	16	20	24	32	36
Initial cell lenght (m)	0.8929	0.7813	0.6250	0.5208	0.3906	0.3472
Max nb of refs. (N_r)	9	9	9	9	9	9
Min cell size (m)	0.01395	0.01220	0.00976	0.00813	0.00610	0.00542
Nb of cells	8.4M	10.2M	14.4M	19.1M	29.2M	35.6M

6.3.3.4 Variations on Reynolds number

This set of computations was done to verify the variation of non-dimensional force as a function of Reynolds number ranging in values from $1{\cdot}10^4$ to $3{\cdot}10^6$. More specifically, a Reynolds number list

$$\{1 \cdot 10^4, 3 \cdot 10^4, 1 \cdot 10^5, 3 \cdot 10^5, 4 \cdot 10^5, 6 \cdot 10^5, 8 \cdot 10^5, 1 \cdot 10^6, 2 \cdot 10^6, 3 \cdot 10^6\}$$

was used. The corresponding inflow velocities are computed by using the Reynolds number definition

$$\mathrm{Re} = \frac{uL_x}{\nu_\mathrm{w}}.$$

The results are shown in Fig. 6.12. From the results it is apparent that non-dimensional force varies with Reynolds number: non-dimensional force increases for lower Reynolds numbers. The medians of the computations were fit into a meta-model where force depends linearly and quadratically on velocity, this means that non-dimensional force has a model

$$C_X = \frac{C_{X_\mathrm{l}}}{\mathrm{Re}} + C_{X_\mathrm{q}}. \tag{6.3}$$

This fit includes only data from computations with Reynolds values up to $1{\cdot}10^6$: this is because VISOR3 does not operate beyond that regime.

Uncertainty related to choosing this meta-model is quantified by using Student's t-distribution uncertainty on the regression error. The resultant fit and confidence interval are depicted in Fig. 6.12. Figure 6.12 includes as well an estimation of the uncertainty of each computation; rather predictably, the model has greater uncertainty for Reynolds greater than $1{\cdot}10^6$.

Finally, in order to further illustrate the model, Fig. 6.13 (a) shows variation on force X as a function of velocity u, when X is computed in Newton and computed from $X = \frac{1}{2}\rho A_{yz} U^2 C_X$.

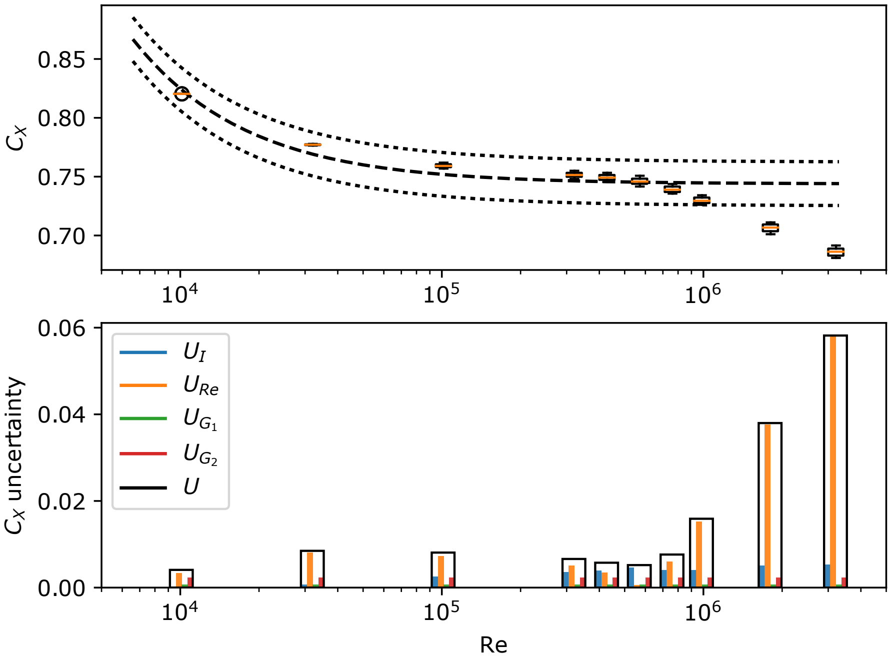


Figure 6.12. Non-dimensional force at different levels of refinement

The model is depicted with (black) dashed lines; the confidence interval of the force is depicted with (black) dotted lines. The depicted confidence interval reunites all sources of uncertainty: domain size, grid refinement, iterative, and Reynolds number. The magnitude of all components, as well as consolidated uncertainty are shown in Fig. 6.13 (b). Total uncertainty is measured as

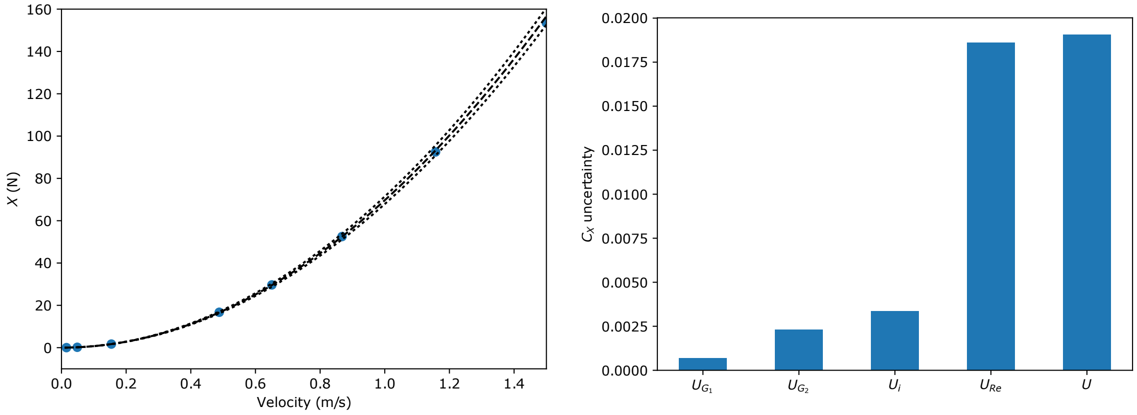
$$U = \sqrt{U_i^2 + U_{G_1}^2 + U_{G_2}^2 + U_{\mathrm{Re}}^2}.$$

From these results, it becomes apparent that the contribution of numerical computation uncertainty to the total uncertainty is small compared to that of assuming model (6.3).

6.3.4 VISCOUS-FLOW COMPUTATIONS

According to Sec. 6.2.3, viscous-flow computations of VISOR3 include the following types of computations:

1. Group 1: steady-state computations of forces and moments for different inflow directions.
2. Group 2: steady-state computations of forces and moments for circular manoeuvres in the XY plane.
3. Group 3: unsteady computations of forces and moments for rotation around the XYZ axes. For roll and pitch motion, oscillations were used; for yaw motion, acceleration



(a)

(b)

Figure 6.13. Non-dimensional force at different velocities

curves were considered.

All computations are run for a total of 2000 iterations. From these iterations, the last 500 are used to obtain the data to be further used. The central value is computed by using the median. From Sec. 6.3.3, uncertainty tends to be dominated first from fitting data to a meta-model and second by iterative uncertainty. Because of this and the fact that the uncertainty analysis was made only for x-direction inflow, uncertainty will be depicted in the plots by using error bars from iterative uncertainty as it was computed in Sec. 6.3.3.

6.3.4.1 Steady-state flow computations at different inflow directions

Steady-state computations of forces and moments for different inflow directions are parametrised by using three-circle designs (3CD), as shown in Sec. 5.4.2. Even though circle designs are a poor choice for meta-model regression (see Sec. 5.4.6), they have the great advantage of allowing one to interpret the data easier, *i.e.*, to understand motion by seeing data around the main planes. All computations within this group were made using a 14M-cell grid with whole external boundary as explained by Fig. 6.5. The circle designs are

- xy-design, given in degrees by

$$q_i = \{0, 10, 15, 20, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180\};$$

- xz-design, given in degrees by

$$q_i = \{-180, -158, -130, -90, -49, -21, 0, 21, 49, 90, 130, 158\}; \quad \text{and}$$

- yz-design, given in degrees by $q_i = \{-90, -49, -21, 0, 21, 49, 90\}$.

For these designs, redundant computations were avoided and xz-plane symmetry was assumed. As it becomes apparent, these designs are not completely uniform: for xy-design the experimental points are chosen heuristically, they are relatively more numerous, and, consequently, their maximin distance is smaller. Figure 6.14 shows a visualisation of some computations of the xy-design. In contrast, xz- and yz-designs are less numerous. Their distribution is apparently arbitrary but it corresponds to a programming error committed when automating the computations. (This error was considered not relevant enough to redo the computations.) It is important to note that xz-design does not take advantage of symmetry but yz-design does.

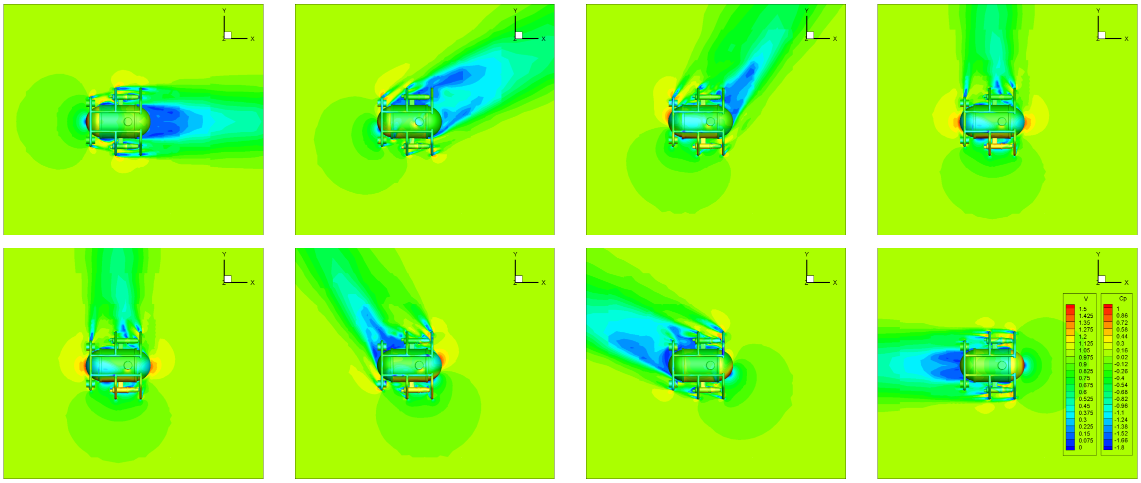
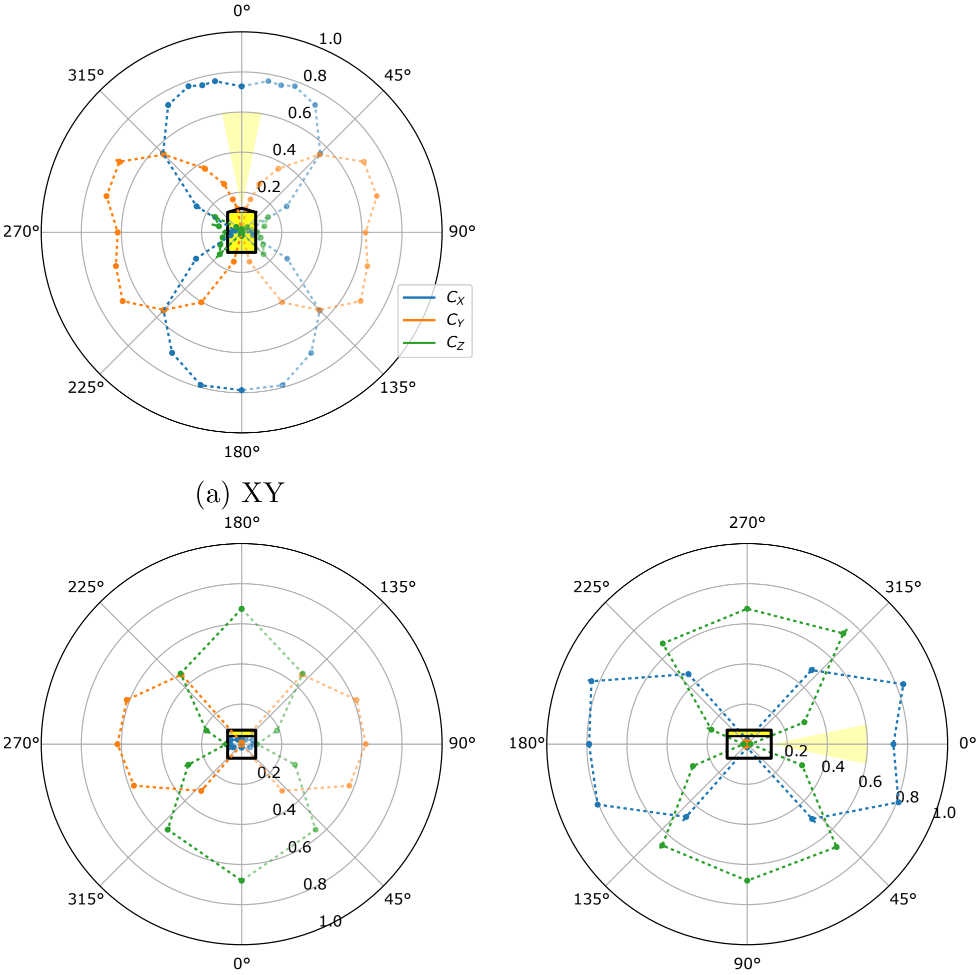


Figure 6.14. Flow visualisation at different inflow drift angles

Figures 6.15 and 6.16 summarise the results obtained for non-dimensional forces and moments from all computations. The main intention behind these polar plots is to illustrate the geometry behind loads and moments behaviour. In the plots, the radius represents the magnitude of the non-dimensional forces and moments according to the definitions in Sec. 6.2.3.1; the plots are done by using the methods proposed in Sec. 4.2. Because xz-plane symmetry was assumed, faded points and lines represent data reflected on the xz-plane. In order to understand the plots more intuitively, all plots depict an ROV that projects light from its front forward.

Figure 6.15 (a), xy-plane plot, represents a top-view slice of the magnitude of non-dimensional forces C_X and C_Y (primarily) as a function of relative velocity direction. This plot exhibits better resolution compared to Figs. 6.15 (b) and (c) due to the way the designs were organised. It is relevant to note the relative magnitude among C_X, C_Y, and C_Z: C_X and C_Y are of



(a) XY

(b) YZ

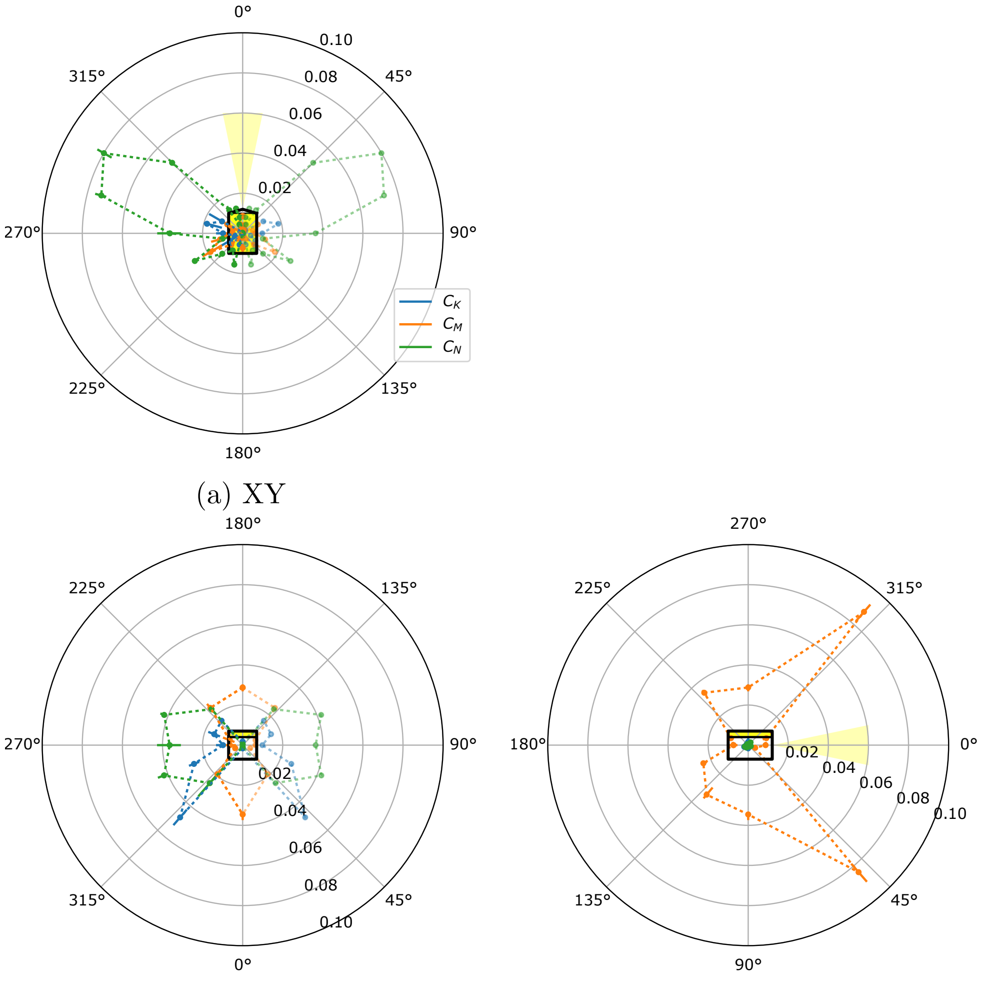
(c) XZ

Figure 6.15. Results for forces at different inflow drift angles

comparable magnitude and C_Z is relatively small. As expected C_X tends to vanish in y-direction and C_Y in x-direction.

Analogous behaviour is evidenced in Figs. 6.15 (b) and (c). Figure 6.15 (b), yz-plane plot, represents a front-view slice that shows primarily the magnitude of non-dimensional forces C_Y and C_Z as a function of relative velocity direction. Figure 6.15 (c), xz-plane plot, represents a side-view slice that shows primarily the magnitude of non-dimensional forces C_X and C_Z as a function of relative velocity direction. In general, forces do not exhibit symmetry with respect xy and yz planes, but assuming such symmetry may be considered reasonable and practical.

Figure 6.16 is Fig. 6.15 non-dimensional moments counterpart. As it becomes apparent, the way moments vary with direction is more complex compared to forces'. It may be conjectured that this is due to VISOR3's relatively complex geometry.



(a) XY

(b) YZ

(c) XZ

Figure 6.16. Results for moments at different inflow drift angles

In order to be more specific, Fig. 6.16 (a), xy-plane plot, represents the top-view slice of the magnitude of non-dimensional moments as a function of relative velocity direction. In xy-plane motion, the most relevant non-dimensional moment component is C_N. This moment is not symmetric with respect to yz-plane and its magnitude is larger when relative velocity goes forward compared to backward. This behaviour, as it is shown in Fig. 6.16 (b) and (c), is replicated analogously in the other planes: C_K in yz-plane and C_M in xz-plane.

6.3.4.2 Steady-state flow computations for circular manoeuvres in the plane

Steady-state computations of forces and moments for circular manoeuvres in the plane were done in the xy-plane exclusively. Because motion is restricted to the xy-plane, direction is parametrised by using a one-circle design (1CD), as shown in Sec. 5.4.2. Conversely, rotation is parametrised by using non-dimensional angular rate (see Sec. 6.2.3.2). These two parameters are combined by using a full-factorial design (FFD) were all combinations of the following parameters were performed:

- 1CD with $q_i = \{0, 30, 60, 90\}$ (in degrees), and
- $\gamma_r = \{0.03, 0.1, 0.3\}$.

This makes a total of 12 different computations. Figure 6.17 shows a visualisation of one of such computations.

Figure 6.18 summarises the results obtained for non-dimensional forces and moments from all computations. As usual, because xz-plane symmetry was assumed, faded points and lines represent data reflected on the xz-plane. These plots show the contour of non-dimensional forces and moments for the different values of non-dimensional yaw angular rate γ_r.

From Fig. 6.18 (a), it becomes apparent that for γ_r from 0 to 0.1 the increase of non-dimensional force is small compared to that of $\gamma_r = 0.3$. Conversely, from Fig. 6.18 (b), it becomes apparent that non-dimensional moment is indeed affected by γ_r, but the effect is less relevant than it is for non-dimensional forces.

6.3.4.3 Unsteady flow computations using harmonic and arbitrary imposed motion in rotation

Unsteady computations of forces and moments for rotation were done around the x, y, and z axes. In this sense, these computations used three-point, zero-dimensional designs as specified in Sec. 5.4.1. More specifically, harmonic oscillations were used for roll and pitch motions; conversely, an acceleration ramp was used for yaw motion. For both roll and pitch motion, the parameters in Table 6.9 were used to define harmonic motion. In order to transition from equilibrium to regular harmonic motion, the amplitude is modulated by a cycloidal function for a number of start-up periods: this type of imposed motion is native to ReFRESCO. Figure 6.19 (a) shows how the imposed angular motion looks like in position and velocity, as well as the resulting computed roll and pitch moments.

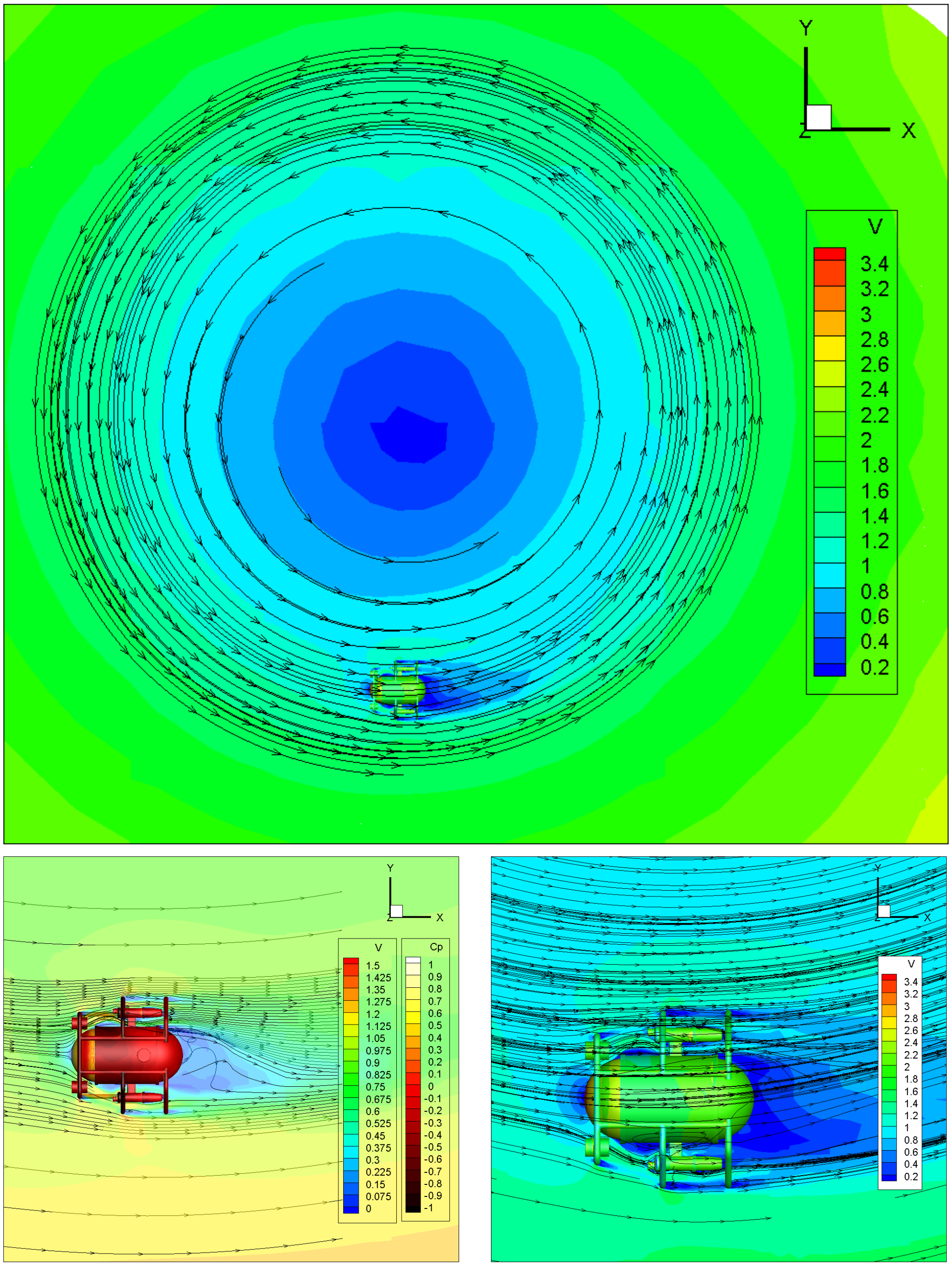
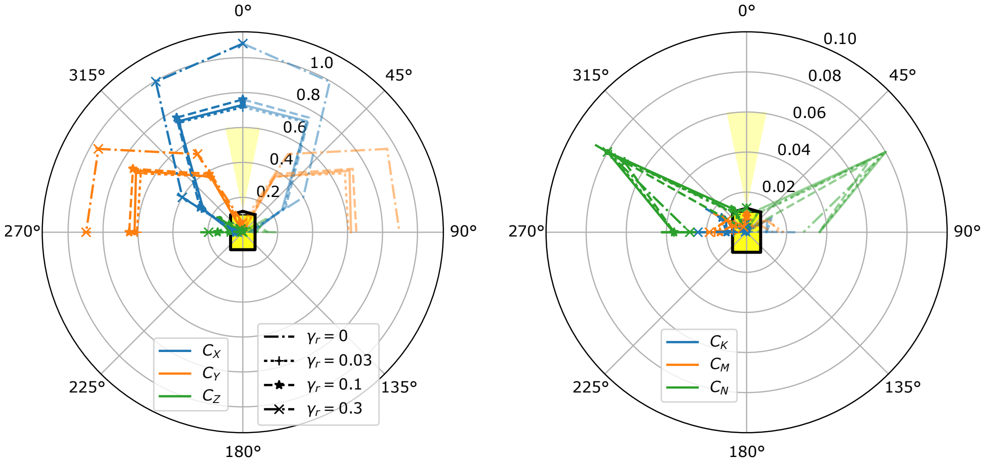


Figure 6.17. Flow visualisation for computations of circular manoeuvres in the xy-plane

For yaw motion, the acceleration ramp was defined by using cycloidal functions that transition from zero to non-zero angular velocity. The acceleration ramp is given for yaw rate r by the function

$$r(t) = \frac{\Delta\psi}{\Delta t}\left(1 - \cos\left(\frac{\pi}{\Delta t}(t - t_0)\right)\right),$$

for $t \in [t_0,\, t_0 + \Delta t]$, where $\Delta\psi$ and Δt are respectively the angle and time used to transition.



(a) (b)

Figure 6.18. Results for forces at different combinations of inflow direction and yaw rotation

Table 6.9. Forced oscillation parameters for roll and pitch

Parameter	Value	Units
Frequency	3.0	rad/s
Amplitude	5	deg
Total number of periods	3	-
Start-up periods	1	-
Time step	0.005	s

In this case, the curve is computed so that after rotating for 90 degrees the ROV has a maximum angular rate at the end of the acceleration ramp of 1.8 rad/s. Figure 6.19 (b) shows the angular position and velocity as a function of time for this experiment, as well as the resulting computed yaw moment.

6.3.5 META-MODEL ASSEMBLY

After obtaining a data-base from viscous-flow computations, it is necessary to define an adequate meta-model structure and then compute a regression to determine its parameters. Computing the meta-model requires both selecting the meta-model's basis functions and computing the coefficients. The selection of the meta-model's basis functions requires computing the regression and obtaining a measure for the uncertainty when a particular set of basis functions are used; consequently, the selected basis functions set is the one that delivers the smallest uncertainty. The process of assembling meta-models is divided according to the type

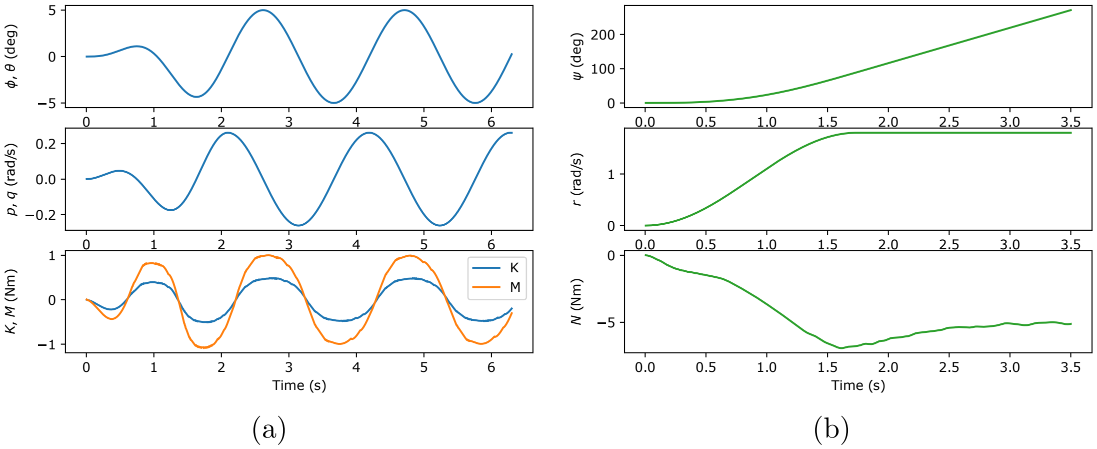


Figure 6.19. Imposed motion: (a) roll and pitch and (b) yaw

of computations pursued previously. In this section, a model for drag due to incoming flow is assembled first and a model for moments due to rotating on a spot is assembled second. These two models are then frozen to finally assemble a model for combined translation and rotation in the horizontal plane.

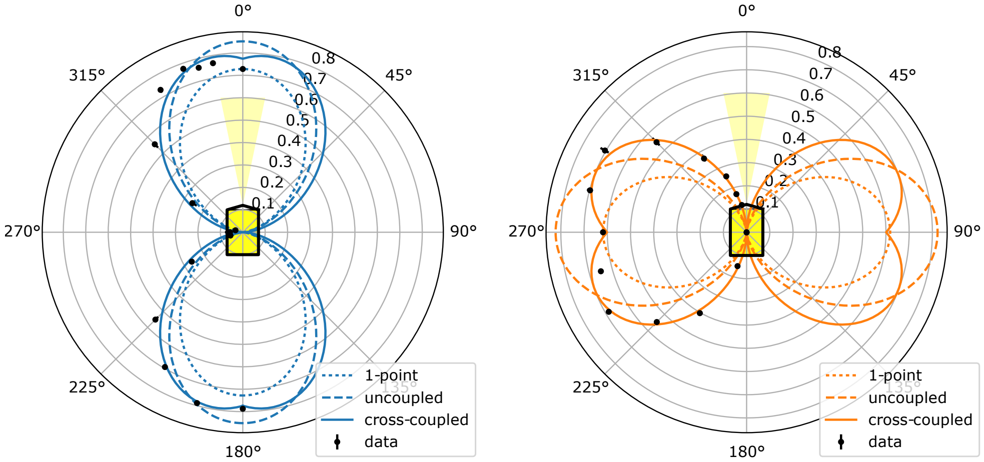
6.3.5.1 A note on choosing appropriate basis functions for the meta-model

A case in point for the adequacy of basis functions to be used in meta-models can be constructed from obtaining non-dimensional forces C_X and C_Y from computations in xy-plane (see Fig. 6.15 (a)). The basis functions are those that allow one to write meta-models such as that of Eq. (6.1), *e.g.*, a basis $\{e_u|e_u|, e_v|e_v|, e_u|e_v|, e_v|e_u|\}$. The results from using different basis and procedures for obtaining the coefficients is summarised in Fig. 6.20.

A first hypothesis is to obtain the model from the minimum information possible, *i.e.*, to obtain C_X from one computation when incoming flow is aligned with x-axis and, analogously, obtain C_Y from one computation when incoming flow is aligned with y-axis. When this idea is extended to C_Z, this type of computer experiment design fits a three-point, zero-dimensional design, as mentioned in Sec. 5.4.1. In this case, the model for non-dimensional forces is

$$C_X(e_u, e_v, e_w) = C_{X,u|u|}e_u|e_u|,$$
$$C_Y(e_u, e_v, e_w) = C_{Y,v|v|}e_v|e_v|.$$

This type of model is uncoupled, because, *e.g.*, C_X does not depend on e_v and vice versa. The result from using this procedure is shown in Fig. 6.20 (a) and (b) as the 1-point graphs. As the plot illustrates, this method is not good at capturing the behaviour of the remaining data; additionally, by using it, drag is under-predicted.



(a) C_X

(b) C_Y

Figure 6.20. Meta-models comparison for (a) C_X and (b) C_Y in the xy-plane

A second hypothesis is to obtain the model from all available data for the uncoupled model used previously. The result from using this procedure is shown in Fig. 6.20 (a) and (b) as the uncoupled graphs. As the plot shows, this model fits the data better, but in some regions over-predicts and in others under-predicts force. More relevantly and from a qualitative perspective, this type of model does not reflect the geometry observed in the data.

A third hypothesis is to add to the meta-model basis cross-coupled terms. This means adding terms where, *e.g.*, C_X depends on e_v. In this case, the meta-model becomes

$$\begin{aligned} C_X(e_u, e_v, e_w) &= C_{X,u|u|} e_u |e_u| + C_{X,u|v|} e_u |e_v|, \\ C_Y(e_u, e_v, e_w) &= C_{Y,v|v|} e_v |e_v| + C_{Y,v|u|} e_v |e_u|. \end{aligned}$$

The result from using this model is shown in Fig. 6.20 (a) and (b) as the cross-coupled graphs. As the plot shows, this model fits the data and reflects the geometry observed in the data better, compared to its competitors.

Besides the uncertainty expected when any meta-model is used, it is important to acknowledge that these types of meta-models assume that loads are symmetrical with respect to all axes. Taking into account that in general forces are not symmetrical, the use of these meta-models will add to the uncertainty. Because this, uncertainty related to choosing one meta-model or another will be discussed next.

6.3.5.2 Assembling a model for drag due to incoming flow

In this section a model for non-dimensional forces and moments due to changing inflow direction is obtained. This model uses the data obtained in Sec. 6.3.4.1 and is intended to predict the behaviour illustrated in Figs. 6.15 and 6.16. This means that regressions are computed from all the data illustrated in these figures. For both forces and moments, the meta-model is chosen from three alternatives based on a measure of uncertainty related to the meta-model. Meta-model uncertainty is measured by computing an error between the data and meta-model prediction, and then obtained by computing a zero-mean 95 % percentile interval from Student's t-distribution. The results from performing the regressions and computing the uncertainties is summarised in Fig. 6.21. The figure shows uncertainty for models 1, 2, and 3 for each force and moment; the models are explained next.

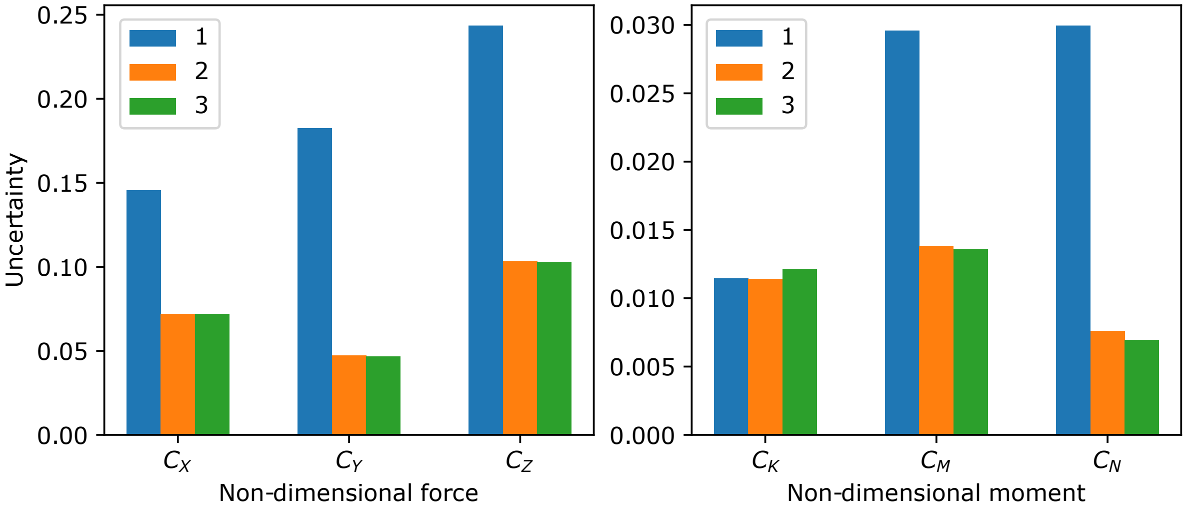


Figure 6.21. Meta-model uncertainty comparison for different types of meta-model

Forces include models obtained from 1) uncoupled, 2) cross-coupled, and 3) linear-quadratic, cross-coupled basis functions. Both uncoupled and cross-coupled basis functions use quadratic functions in modulo form.

- Model 1 is obtained from an uncoupled basis where C_X only depends on e_u and so on. The model for each force is written as

$$\begin{aligned} C_X(e_u, e_v, e_w) &= C_{X,u|u|} e_u |e_u| + \cdots , \\ C_Y(e_u, e_v, e_w) &= C_{Y,v|v|} e_v |e_v| + \cdots , \\ C_Z(e_u, e_v, e_w) &= C_{Z,w|w|} e_w |e_w| + \cdots . \end{aligned}$$

  This model is the simplest but, from the results Fig. 6.21, is the one that shows more uncertainty. Please note that this is the type of model used in Minerva (see Sec. 3.3). In general, and acknowledging the discussion in the previous section, it is not a good choice for the model.

- Model 2 is obtained from a cross-coupled basis where, for instance, C_X, besides of its dependence on e_u, depends on e_v and e_w as well. In this case, the model for each force is written as

$$\begin{aligned} C_X(e_u, e_v, e_w) &= C_{X,u|u|} e_u |e_u| + C_{X,u|v|} e_u |e_v| + C_{X,u|w|} e_u |e_w| + \cdots\,, \\ C_Y(e_u, e_v, e_w) &= C_{Y,v|u|} e_v |e_u| + C_{Y,v|v|} e_v |e_v| + C_{Y,v|w|} e_v |e_w| + \cdots\,, \\ C_Z(e_u, e_v, e_w) &= C_{Z,w|u|} e_w |e_u| + C_{Z,w|v|} e_w |e_v| + C_{Z,w|w|} e_w |e_w| + \cdots\,. \end{aligned}$$

  this type of model is less simple but, from the results Fig. 6.21, it improves the uncertainty greatly; consequently, this is a better choice for the model.

- Model 3 is the same as the previous one but includes a linear component; consequently, the model for each force is

$$\begin{aligned} C_X(e_u, e_v, e_w) &= C_{X,u} e_u + C_{X,u|u|} e_u |e_u| + C_{X,u|v|} e_u |e_v| + C_{X,u|w|} e_u |e_w| + \cdots\,, \\ C_Y(e_u, e_v, e_w) &= C_{Y,v} e_v + C_{Y,v|u|} e_v |e_u| + C_{Y,v|v|} e_v |e_v| + C_{Y,v|w|} e_v |e_w| + \cdots\,, \\ C_Z(e_u, e_v, e_w) &= C_{Z,w} e_w + C_{Z,w|u|} e_w |e_u| + C_{Z,w|v|} e_w |e_v| + C_{Z,w|w|} e_w |e_w| + \cdots\,. \end{aligned}$$

  From the results in Fig. 6.21, this model makes an improvement in the uncertainty, but the improvement is small.

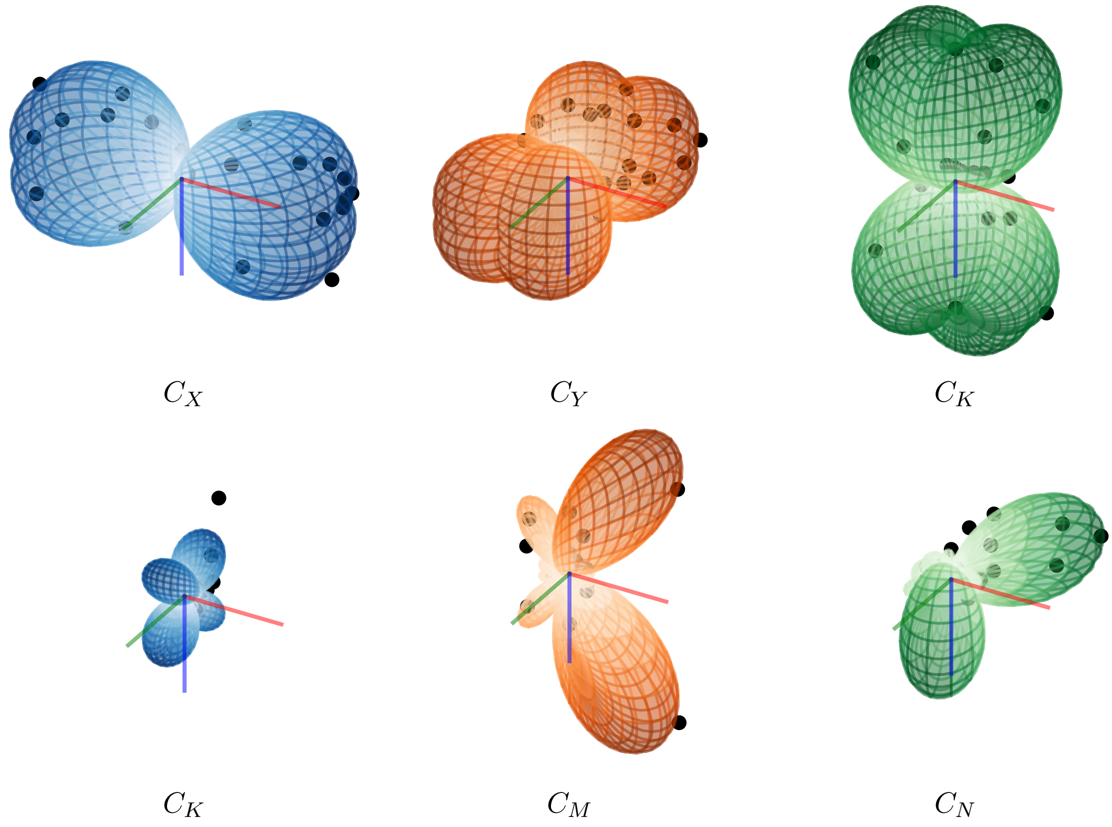
From the models at hand, Fig. 6.22 illustrates the geometry of the selected meta-model for each force, as well as how the model represents the data points. Based on the information in Fig. 6.21, Model 2 is selected as the best option for all forces.

Moments include models obtained from second-order, third-order, and square-root basis functions.

- Model 1 is obtained from a second-order basis with a quadratic term in modulo form and a regular cross-coupled term. The model for each moment is written as

$$\begin{aligned} C_K(e_u, e_v, e_w) &= C_{K,w|w|} e_w |e_w| + C_{K,vw} e_v e_w + \cdots\,, \\ C_M(e_u, e_v, e_w) &= C_{M,w|w|} e_w |e_w| + C_{M,uw} e_u e_w + \cdots\,, \\ C_N(e_u, e_v, e_w) &= C_{N,v|v|} e_v |e_v| + C_{N,uv} e_u e_v + \cdots\,. \end{aligned}$$

  Taking into account the results in Fig. 6.21, the behaviour of uncertainty is different depending on the particular moment. For C_M and C_N, it shows the worst uncertainty metric, but for C_K the behaviour is comparable to that of other meta-models.



C_X C_Y C_K

C_K C_M C_N

Figure 6.22. Meta-models' three-dimensional illustration

- Model 2 adds a third-order term that captures the variation observed in the moment better. This will become rather evident from Figs. 6.24, 6.25 and 6.26. In this case, the model for each force is written as

$$
\begin{aligned}
C_K(e_u, e_v, e_w) &= C_{K,w|w|} e_w |e_w| + C_{K,vw} e_v e_w + C_{K,vw^2} e_v e_w |e_w| + \cdots \,, \\
C_M(e_u, e_v, e_w) &= C_{M,w|w|} e_w |e_w| + C_{M,uw} e_u e_w + C_{M,uw^2} e_u e_w |e_w| + \cdots \,, \\
C_N(e_u, e_v, e_w) &= C_{N,v|v|} e_v |e_v| + C_{N,uv} e_u e_v + C_{N,uv^2} e_u e_v |e_v| + \cdots \,.
\end{aligned}
$$

  From the point of view of uncertainty, these models show acceptable behaviour compared to its competitors.
- Model 3 adds an additional term that uses a square-root basis function. This component looks forward to capture non-symmetry of the moments. By using this term, the model

for each moment becomes

$$
\begin{aligned}
C_K(e_u, e_v, e_w) =& C_{K,w|w|} e_w |e_w| + C_{K,vw} e_v e_w + C_{K,vw^2} e_v e_w |e_w| + \\
& C_{K,vw^2} (v^2 - w^2) e_v e_w \sqrt{\tfrac{1}{2}(1-v)} + \cdots , \\
C_M(e_u, e_v, e_w) =& C_{M,w|w|} e_w |e_w| + C_{M,uw} e_u e_w + C_{M,uw^2} e_u e_w |e_w| + \\
& C_{K,vw^2} (u^2 - w^2) e_u e_w \sqrt{\tfrac{1}{2}(1-w)} + \cdots , \\
C_N(e_u, e_v, e_w) =& C_{N,v|v|} e_v |e_v| + C_{N,uv} e_u e_v + C_{N,uv^2} e_u e_v |e_v| + \\
& C_{K,vw^2} (u^2 - v^2) e_u e_v \sqrt{\tfrac{1}{2}(1-u)} + \cdots .
\end{aligned}
$$

From the results Fig. 6.21, there is not a clear improvement from using this basis function.

As it was done with the forces, Fig. 6.22 illustrates also the geometry of the selected meta-model for each moment, as well as how the model represents the data points. For C_K Model 1 was used; conversely, for C_M and C_N Model 2 was used.

Figure 6.23 illustrates uncertainty of the meta-model selected for each force. All plots show iterative, meta-model, and total uncertainties: respectively, U_i, U_{mm}, and U. As usual, total uncertainty is computed from Euclidean norm. These results suggest that between iterative and meta-model uncertainty the more relevant is meta-model uncertainty.

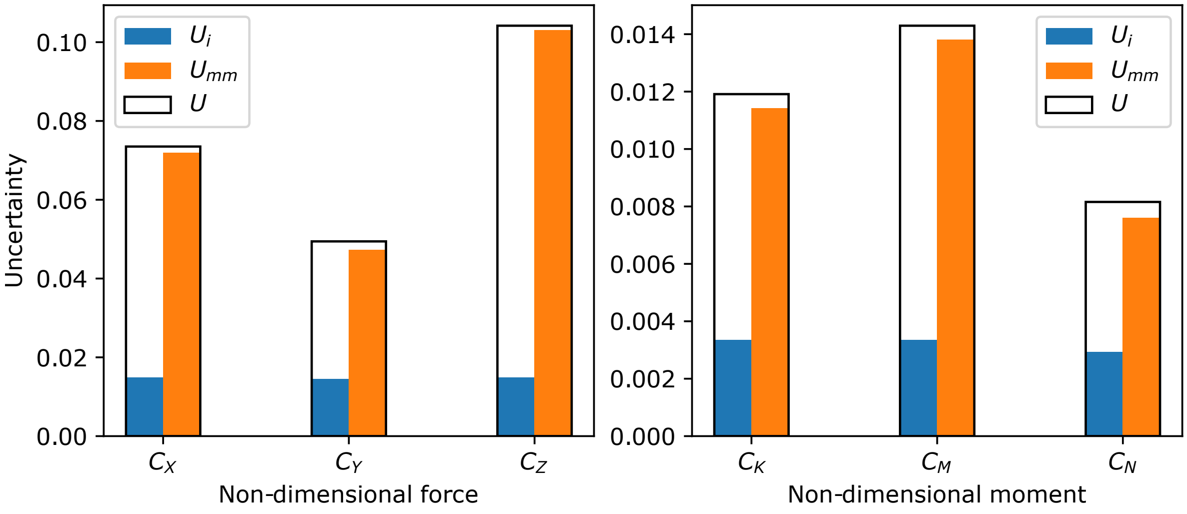


Figure 6.23. Uncertainty of the selected meta-model

Finally, Figs. 6.24, 6.25, and 6.26 further illustrate data and meta-models in a two-dimensional, Cartesian representation. These plots show data-points as dots and meta-model as a solid line; furthermore, the interval given by total uncertainty is depicted by dotted lines. In Fig. 6.24

an xy-plane illustration is shown. In this plot, forces and moments are plotted against a drift angle β, which is computed as $\beta = \arctan2(e_u, e_v)$. To better interpret this angle, zero degrees represent relative velocity along positive x-axis and 180 degrees along negative x-axis. Conversely, 90 degrees represent relative velocity along positive y-axis and 180 degrees around negative y-axis. In this case, relative velocity along z-axis is zero.

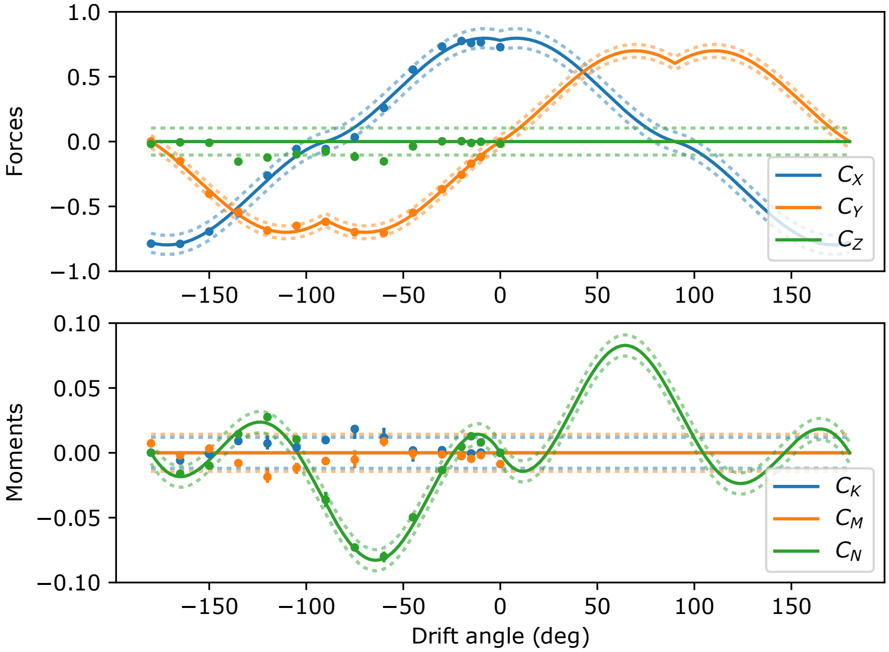


Figure 6.24. Meta-model curves in cartesian representation: xy-plane

In Fig. 6.25 an xz-plane illustration is shown. In this plot, forces and moments are plotted against an attack angle α, which is computed as $\alpha = \arctan2(e_w, e_u)$. As it was done previously, to better interpret α, zero degrees represent relative velocity along positive x-axis and 180 degrees along negative x-axis, similarly as in the xy-plane plots. On the other hand, in this case, 90 degrees represent relative velocity along positive z-axis (downward relative motion) and 180 degrees around negative z-axis (upward relative motion). In this case, relative velocity along y-axis is zero.

In Fig. 6.26 a yz-plane illustration is shown. In this plot, forces and moments are plotted against a drift angle β, which is computed as $\alpha = \arctan2(e_v, e_w)$. As usual, to better interpret this drift angle, zero degrees represent relative velocity along positive z-axis (downward relative motion) and 180 degrees along negative z-axis (upward relative motion). Then, 90 degrees represent relative velocity along positive y-axis and 180 degrees around negative y-axis . In all cases relative velocity along x-axis is zero.

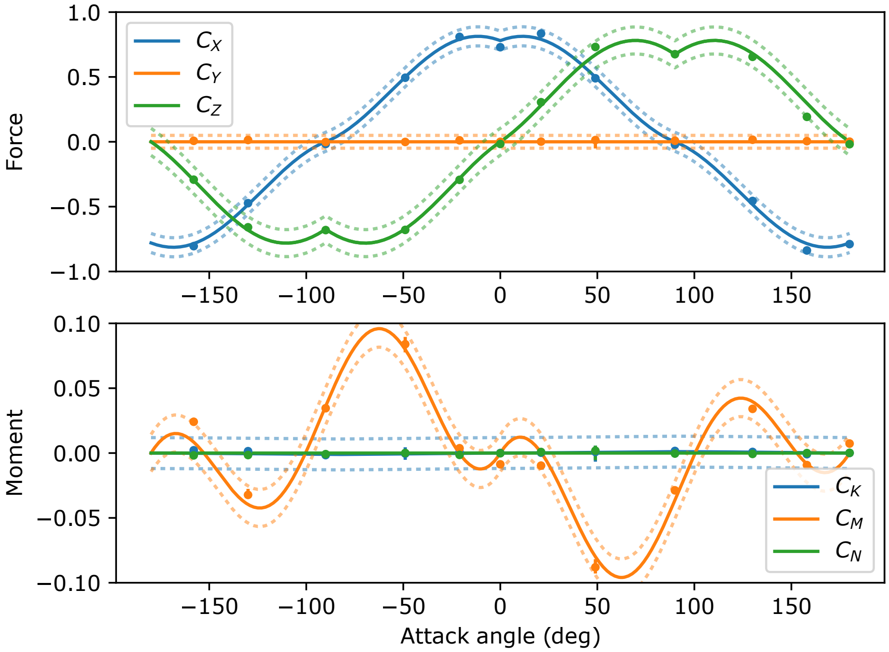


Figure 6.25. Meta-model curves in cartesian representation: xz-plane

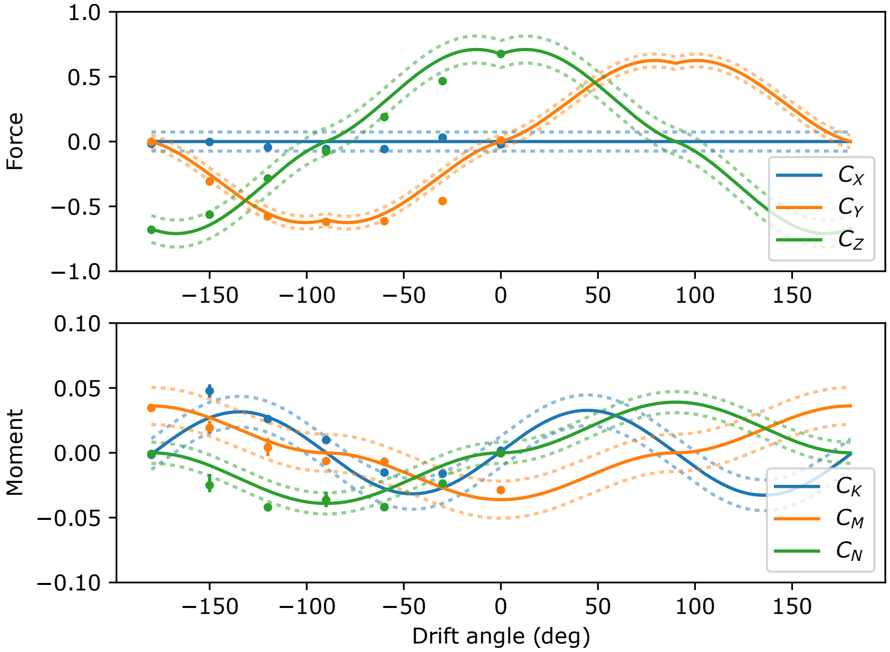


Figure 6.26. Meta-model curves in cartesian representation: yz-plane

The drag forces model is finally

$$C_X(e_u, e_v, e_w) = C_{X,u|u|}e_u|e_u| + C_{X,u|v|}e_u|e_v| + C_{X,u|w|}e_u|e_w| + \cdots, \tag{6.4}$$

$$C_Y(e_u, e_v, e_w) = C_{Y,v|u|}e_v|e_u| + C_{Y,v|v|}e_v|e_v| + C_{Y,v|w|}e_v|e_w| + \cdots, \tag{6.5}$$

$$C_Z(e_u, e_v, e_w) = C_{Z,w|u|}e_w|e_u| + C_{Z,w|v|}e_w|e_v| + C_{Z,w|w|}e_w|e_w| + \cdots. \tag{6.6}$$

Conversely, the drag moments model is

$$C_K(e_u, e_v, e_w) = C_{K,w|w|} e_w |e_w| + C_{K,vw} e_v e_w + \cdots , \tag{6.7}$$

$$C_M(e_u, e_v, e_w) = C_{M,w|w|} e_w |e_w| + C_{M,uw} e_u e_w + C_{M,uw^2} e_u e_w |e_w| + \cdots , \tag{6.8}$$

$$C_N(e_u, e_v, e_w) = C_{N,v|v|} e_v |e_v| + C_{N,uv} e_u e_v + C_{N,uv^2} e_u e_v |e_v| + \cdots . \tag{6.9}$$

For the models the coefficients obtained from the regression are

- $C_{X,u|u|} = 0.78$, $C_{X,u|v|} = 0.23$, $C_{X,u|w|} = 0.33$.
- $C_{Y,v|u|} = 0.52$, $C_{Y,v|v|} = 0.60$, $C_{Y,v|w|} = 0.24$.
- $C_{Z,w|u|} = 0.58$, $C_{Z,w|v|} = 0.32$, $C_{Z,w|w|} = 0.67$.
- $C_{K,w|w|} = 0.001$, $C_{K,vw} = 0.064$.
- $C_{M,w|w|} = -0.036$, $C_{M,uw} = 0.137$, $C_{M,uw^2} = -0.340$.
- $C_{N,v|v|} = 0.039$, $C_{N,uv} = -0.141$, $C_{N,uv^2} = 0.302$.

6.3.5.3 Assembling a model for moments due to rotating on a spot

In this section a model for moments due to rotating on a spot is obtained. This model uses the data obtained in Sec. 6.3.4.3 and is intended to predict the behaviour shown in Fig. 6.19. As mentioned, roll and pitch come from imposed harmonic oscillation and yaw from a smooth acceleration ramp. Harmonic imposed motion consists of three rotation periods from which the first one is a start-up period. The smooth acceleration ramp uses cycloidal functions. In order to compute curve-fit regression for roll and pitch, only data from the two last periods is used. Conversely, for yaw's acceleration ramp, only data from half a revolution is used, *i.e.*, angles from 0 to 180 degrees: this is done so the ROV does not encounter its own wake.

The meta-model is written in terms of manoeuvring coefficients as

$$\begin{aligned} K &= K_p p + K_{p|p|} p|p| + K_{\dot{p}} \dot{p}, \\ M &= M_q q + M_{q|q|} q|q| + M_{\dot{q}} \dot{q}, \\ N &= N_r r + N_{r|r|} r|r| + N_{\dot{r}} \dot{r}. \end{aligned}$$

The regression is computed from a least-squares curve fit with restrictions in the coefficients, so that only positive coefficients are obtained. From this restriction, coefficients K_p, M_q, and N_r are zero. The fitted curves along the data are shown in Fig. 6.27 The figure includes as well a measure of meta-model uncertainty computed from the error between the data and meta-model prediction. As usual, uncertainty is obtained by computing a zero-mean 95 % percentile interval from Student's t-distribution.

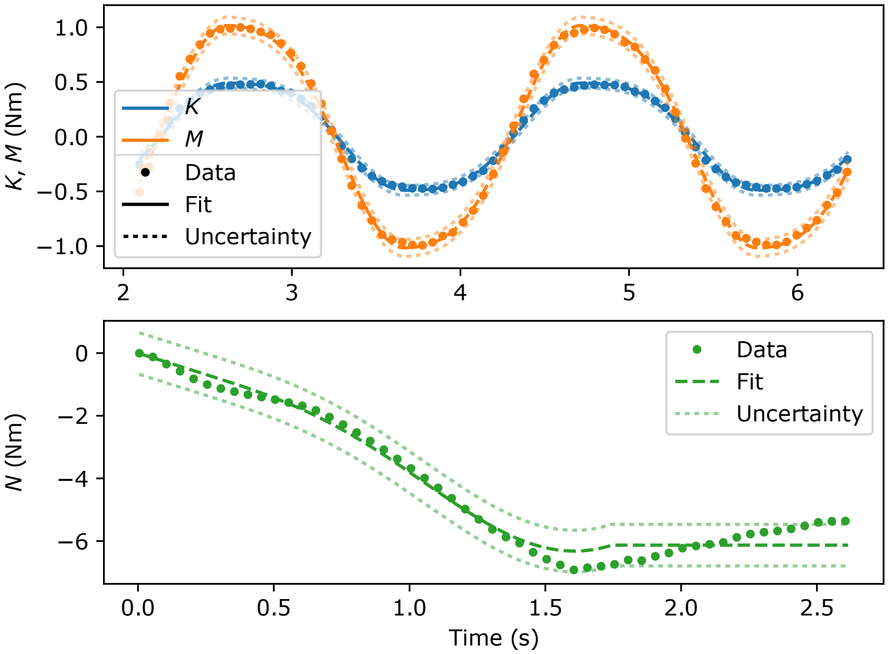


Figure 6.27. Meta-model curves for rotating on a spot

Once the manoeuvring coefficients are known it is possible to compute non-dimensional moments according to the formulation in Sec. 6.2.3.3. More explicitly, the coefficients are computed as

$$\begin{aligned}
C_{K,p|p|} &= \frac{K_{p|p|}}{\frac{1}{2}\rho A_{yz} L_x^3}, & C_{M,q|q|} &= \frac{M_{q|q|}}{\frac{1}{2}\rho A_{xz} L_y^3}, & C_{N,r|r|} &= \frac{N_{r|r|}}{\frac{1}{2}\rho A_{xy} L_z^3}, \\
C_{K,\dot{p}} &= \frac{K_{\dot{p}}}{\rho \nabla L_x^2}, & C_{M,\dot{q}} &= \frac{M_{\dot{q}}}{\rho \nabla L_y^2}, & C_{N,\dot{r}} &= \frac{N_{\dot{r}}}{\rho \nabla L_z^2};
\end{aligned}$$

Finally, the coefficients are

- $C_{K,p|p|} = 0.114$, $C_{M,q|q|} = 0.120$, $C_{N,r|r|} = 0.041$,
- $C_{K,\dot{p}} = 0.023$, $C_{M,\dot{q}} = 0.042$, $C_{N,\dot{r}} = 0.032$.

6.3.5.4 Assembling a model for combined translation and rotation in the horizontal plane

In this section a model for forces and moments due to combined translation and rotation in the horizontal plane is obtained. This model uses the data obtained in Sec. 6.3.4.2 and is intended to predict the behaviour shown in Fig. 6.18. The meta-model considered here includes non-dimensional forces and moments C_X, C_Y, and C_N, as it is customary in horizontal plane motion. The two-basis-function meta-model considered for these loads is

$$\begin{aligned}
C_X(e_u, e_v, \gamma_r) &= \cdots + C_{X,u|r|} e_u |\gamma_r| + C_{X,vr} e_v \gamma_r + \cdots \,, \\
C_Y(e_u, e_v, \gamma_r) &= \cdots + C_{Y,v|r|} e_v |\gamma_r| + C_{Y,ur} e_u \gamma_r + \cdots \,, \\
C_N(e_u, e_v, \gamma_r) &= \cdots + C_{N,|v|r} |e_v| \gamma_r + C_{N,|u|r} |e_u| \gamma_r + \cdots \,.
\end{aligned}$$

The selection of the meta-model consists of selecting from using the first basis function only, namely Model 1, or the two basis functions, namely Model 2. Because in Secs. 6.3.5.2 and 6.3.5.3 models for C_X, C_Y, and C_N were already approached, the regressions computed in this section are done from the residuals between the data in Fig. 6.18 and the predictions of Eqs. (6.4)–(6.9).

As usual, the selection between Model 1 and 2 is based on measured uncertainty by computing a zero-mean 95 % percentile interval from Student's t-distribution. After computing the required regressions, three measures of uncertainty are depicted in Fig. 6.28. Model 0 uncertainty is computed from the error between the data in Fig. 6.18 and the predictions of Eqs. (6.4)–(6.9), *i.e.*, if Models 1 and 2 are not considered. This uncertainty is computed for all non-dimensional forces and moments and serves as baseline for evaluating how the model gets improved by further regressions. Model 1 and 2 uncertainties are computed from the error between the data in Fig. 6.18 and the residuals after the corresponding regression. Additionally, Fig. 6.28 includes a dotted black line that indicates the uncertainty computed in Sec 6.3.5.2 and shown in Fig. 6.23. This illustrates how uncertainty is bigger when this new set of data is included.

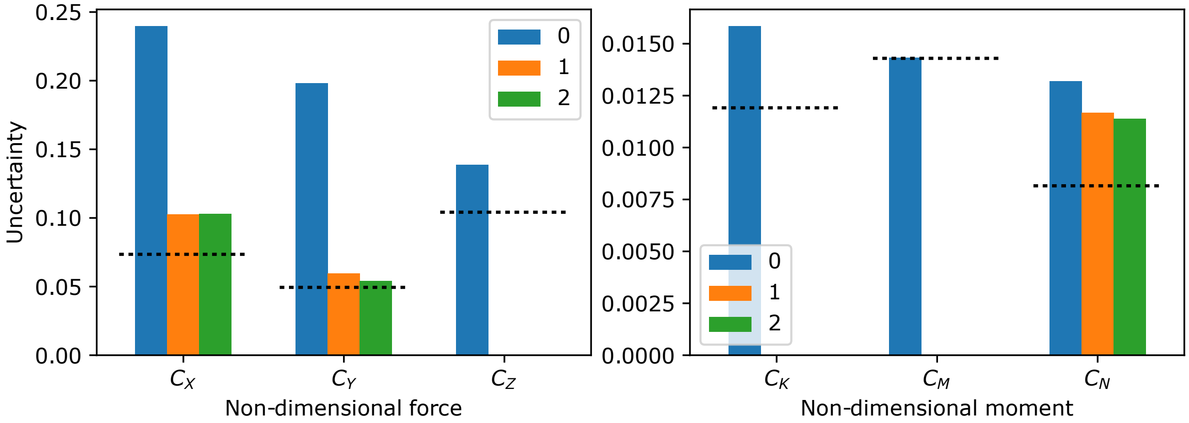


Figure 6.28. Uncertainty of the different meta-models for combined translation and rotation in the horizontal plane

From the results in Fig. 6.28, Model 1 is selected for C_X and Model 2 for C_Y and C_N.

Consequently the model for combined rotation and translation in the horizontal plane becomes

$$C_X(e_u, e_v, \gamma_r) = \cdots + C_{X,u|r|} e_u |\gamma_r| + \cdots \,, \tag{6.10}$$

$$C_Y(e_u, e_v, \gamma_r) = \cdots + C_{Y,v|r|} e_v |\gamma_r| + C_{Y,ur} e_u \gamma_r + \cdots \,, \tag{6.11}$$

$$C_N(e_u, e_v, \gamma_r) = \cdots + C_{N,|v|r} |e_v| \gamma_r + C_{N,|u|r} |e_u| \gamma_r + \cdots \,. \tag{6.12}$$

In order to illustrate this model, Fig. 6.29 shows a Cartesian representation similar to that of Fig. 6.24 where C_X, C_Y, and C_N are plotted against drift angle $\beta = \text{arctan2}(e_u, e_v)$; accordingly, Fig. 6.29 shows C_X, C_Y, and C_N against drift angle for $\gamma_r = 0$ (no rotation) and $\gamma_r = 0.3$. Particularly, the figure shows how the model for C_X and C_Y fit to the increment of these loads when non-dimensional yaw rate increases to $\gamma_r = 0.3$; it shows as well how C_N is modified for $\gamma_r = 0.3$. Finally, after computing the regression, the coefficients are

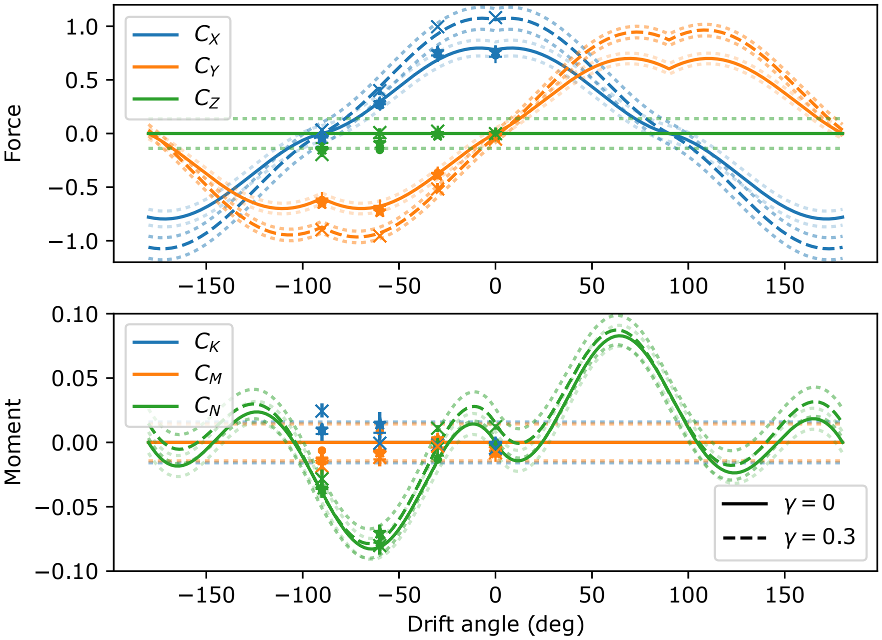


Figure 6.29. Meta-model curves for combined translation and rotation in the horizontal plane

- $C_{X,u|r|} = 0.93$,
- $C_{Y,v|r|} = 0.90$, $C_{Y,ur} = -0.10$,
- $C_{N,|v|r} = 0.036$, $C_{N,|u|r} = -0.016$.

6.3.5.5 Complete meta-model

The results from the previous sections are combined here to write a complete meta-model for hydrodynamic forces and moments in component form. The model is written in terms of full dimensional loads, is dependent on full-dimensional velocities and accelerations, and uses non-dimensional loads' coefficients. The model is then given by

$$\begin{aligned}
X =&\tfrac{1}{2}\rho A_{yz}\left(C_{X,u|u|}u|u| + C_{X,u|v|}u|v| + C_{X,u|w|}u|w| + L_x C_{X,u|r|}u|r|\right), && (6.13)\\
Y =&\tfrac{1}{2}\rho A_{xz}\left(C_{Y,v|u|}v|u| + C_{Y,v|v|}v|v| + C_{Y,v|w|}v|w| + L_y C_{Y,v|r|}v|r| + L_y C_{Y,ur}ur\right), && (6.14)\\
Z =&\tfrac{1}{2}\rho A_{xy}\left(C_{Z,w|u|}w|u| + C_{Z,w|v|}w|v| + C_{Z,w|w|}w|w|\right). && (6.15)\\
K =&\tfrac{1}{2}\rho A_{yz}L_x\left(C_{K,w|w|}w|w| + C_{K,vw}vw + L_x^2 C_{K,p|p|}p|p|\right) + \tfrac{1}{2}\rho\nabla L_x^2 C_{K,\dot{p}}\dot{p}, && (6.16)\\
M =&\tfrac{1}{2}\rho A_{xz}L_y\left(C_{M,w|w|}w|w| + C_{M,uw}uw + C_{M,uw^2}uw|w| + L_y^2 C_{M,q|q|}q|q|\right) +\\
&\tfrac{1}{2}\rho\nabla L_y^2 C_{M,\dot{q}}\dot{q}, && (6.17)\\
N =&\tfrac{1}{2}\rho A_{xy}L_z\left(C_{N,v|v|}v|v| + C_{N,uv}uv + C_{N,uv^2}uv|v| +\right.\\
&\qquad\left. L_z C_{N,|v|r}|v|r + L_z C_{N,|u|r}|u|r + L_z^2 C_{N,r|r|}r|r|\right) + \tfrac{1}{2}\rho\nabla L_z^2 C_{N,\dot{r}}\dot{r}. && (6.18)
\end{aligned}$$

This model could be straight-forwardly written in terms of manoeuvring coefficients as well.

6.4 PRELIMINARY TIME-DOMAIN SIMULATIONS OF VISOR3

In order to test the time-domain meta-model for hydrodynamic forces and moments, two types of preliminary time-domain simulations that do not involve thrusters are computed, namely free drift and impulse responses. The former evaluates when the ROV drifts because of incoming current in the horizontal plane; the latter evaluates when the different degrees of freedom are excited and energy dissipates due to interaction with the fluid. These simulations are preliminary because they are intended to evaluate the obtained hydrodynamics meta-model without much ado about time-domain simulation: these topics will be further dealt with in Chapters 8 and 9. However, the simulations presented in this section are numerical solutions of model (2.5) when the right-hand side of the equation is zero (*i.e.*, external forces are zero), the particulars of VISOR3's model, as explained in Sec. 3.1, are taken into account, and the hydrodynamic model given by Eqs. (6.13)–(6.18) is used.

6.4.1 FREE DRIFT

Free drift motion is computed by simulating constant current that comes from four different directions, namely from current direction angles of 5, 30, 60, and 85 deg (see Fig. 6.30). Incoming current magnitude is 0.2 m/s, representing low-magnitude, near-calm current. The simulation time window is 60 s. The resultant computed trajectories are illustrated in Fig. 6.30. Because

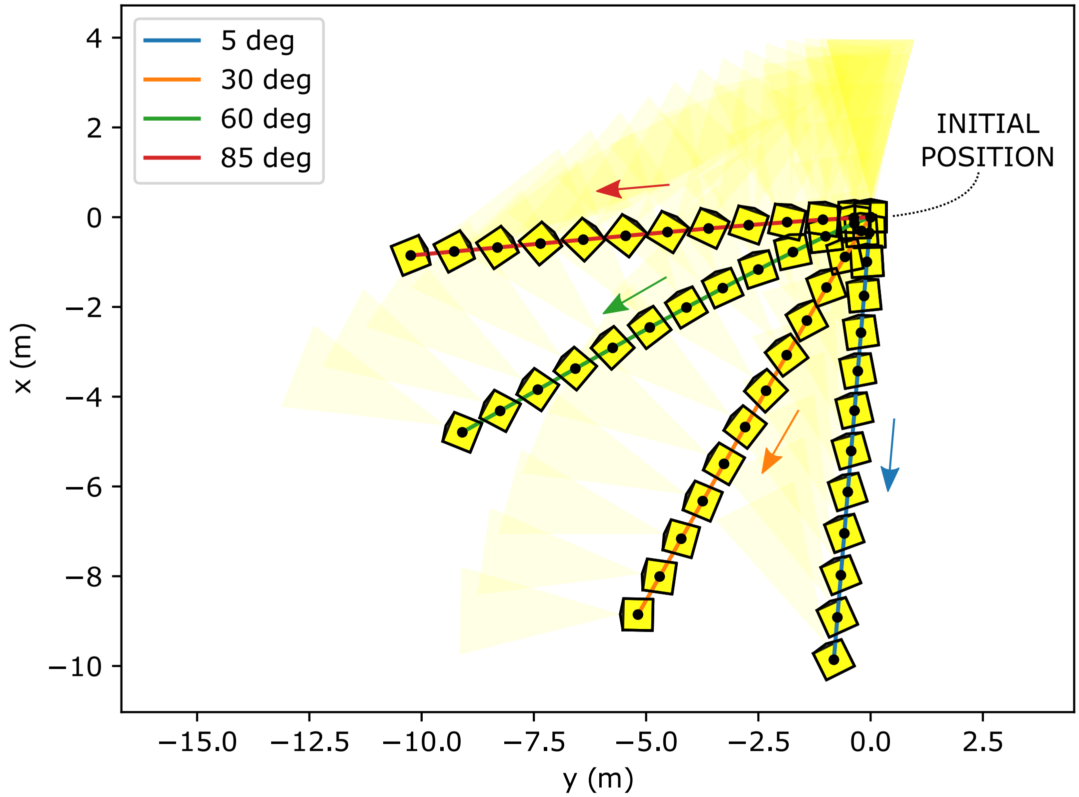


Figure 6.30. Free drift trajectories

of free drift, the ROV's overall motion direction is determined by flow direction. Nevertheless, it can be noticed how the ROV is not able to compensate for yaw moment naturally: it does not keep its heading while drifting. This is expected from the behaviour of C_N where incoming flow at different angles produces yaw moment (see Fig. 6.16); consequently, this moment changes the ROV's heading over time. These results suggest that the ROV requires active yaw control to compensate for cross-flow-induced yaw moment.

6.4.2 IMPULSE RESPONSES

Impulse response simulation is done by exciting all six degrees of freedom individually. The main intention is to show how energy dissipates when the particular degree of freedom is excited and how it is coupled with other degrees of freedom. To check behaviour of displacement degrees of freedom, velocities u, v, and w are plotted with respect to time; conversely, to check behaviour of rotational degrees of freedom, Euler angles ϕ, θ, and ψ are plotted with respect to time in most cases (with the exception of yaw).

Surge impulse response is done by defining an initial surge velocity (u) of 0.5 m/s and simulating a time window of 30 s. The results are shown in Fig. 6.31. The figure shows how u dissipates according to quadratic drag. A relevant behaviour is the way pitch is excited by surge motion, compared to any other degree of freedom. This suggests that pitch oscillation frequency should be taken into account when designing forward motion controllers.

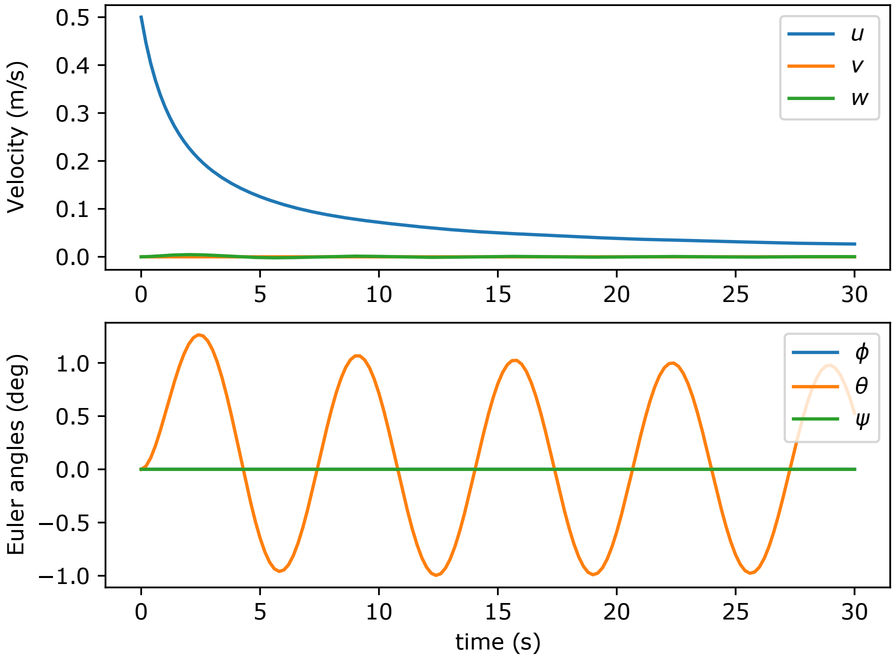


Figure 6.31. Surge impulse response

Sway impulse response is done by defining an initial sway velocity (v) of 0.5 m/s and simulating a time window of 30 s. The results are shown in Fig. 6.32. The figure shows the dissipation of v according to quadratic drag. A relevant behaviour is the way all rotational degrees of freedom are excited by sway motion. Besides roll and pitch oscillations, sway motion is accompanied by a change of heading; this is evidenced by the way ψ goes of the chart in Fig. 6.32. This suggests that pure sway motion requires active control of heading.

Heave impulse response is done by defining an initial heave velocity (w) of 0.5 m/s and simulating a time window of 30 s. The results are shown in Fig. 6.33. As in the previous cases, the figure shows the dissipation of w according to quadratic drag. Similarly as in surge, pitch is excited by heave motion and is more relevant to this degree of freedom that to any other. This suggests that heave motion gets benefited by considering pitch oscillation frequency during control design.

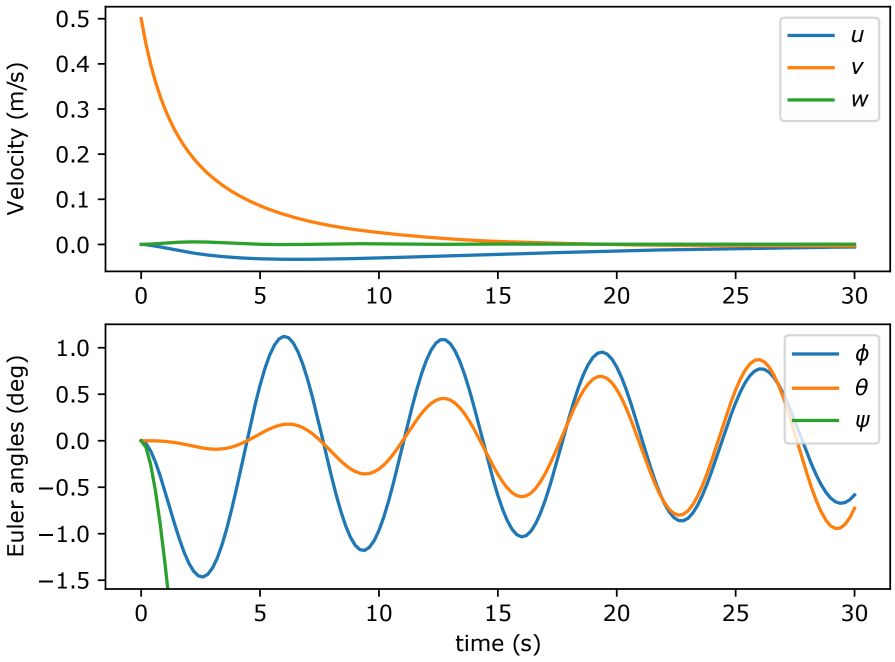


Figure 6.32. Sway impulse response

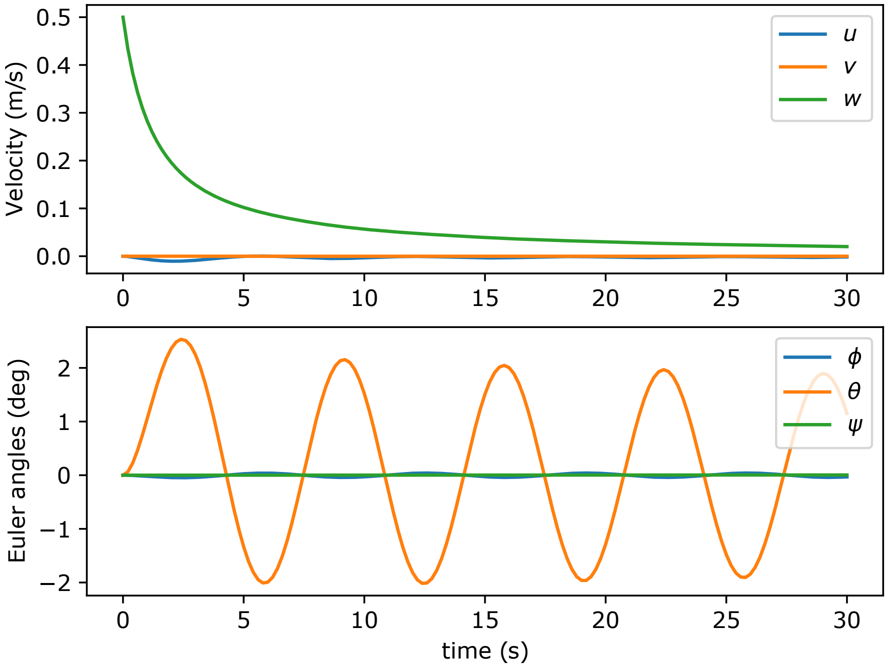


Figure 6.33. Heave impulse response

Roll impulse response is done by defining an initial roll angle (ϕ) of 15 deg and simulating a time window of 30 s. The results are shown in Fig. 6.34. The figure shows the dissipation

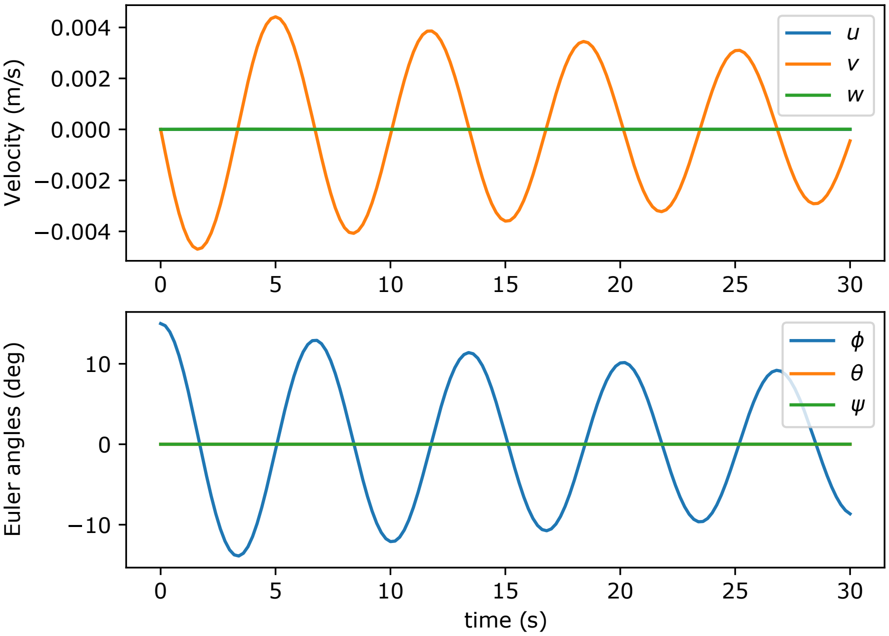


Figure 6.34. Roll impulse response

of ϕ according to quadratic roll moment. A relevant behaviour is the way sway gets slightly excited by roll motion. Pitch impulse response is done by defining an initial pitch angle (θ) of

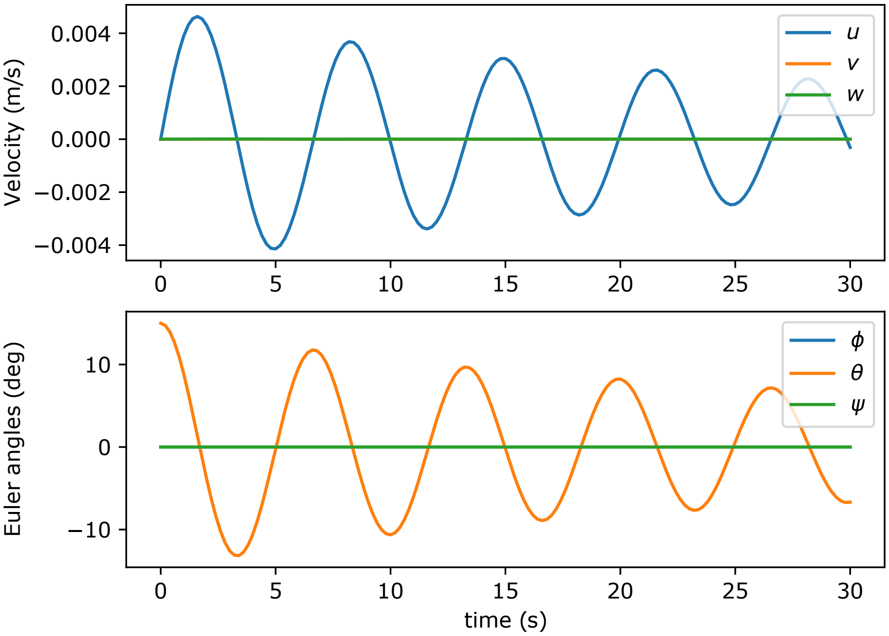


Figure 6.35. Pitch impulse response

15 deg and simulating a time window of 30 s. The results are shown in Fig. 6.35. The figure shows the dissipation of θ according to quadratic pitch moment. A relevant behaviour is the way surge gets slightly excited by roll motion.

Yaw impulse response is done by defining an initial yaw angular velocity (r) of 0.5 rad/s and simulating a time window of 120 s. The results are shown in Fig. 6.36. The figure shows how r dissipates according to quadratic yaw moment and evidences the speed of convergence. In this case, excitation of yaw does not excite any other degree of freedom.

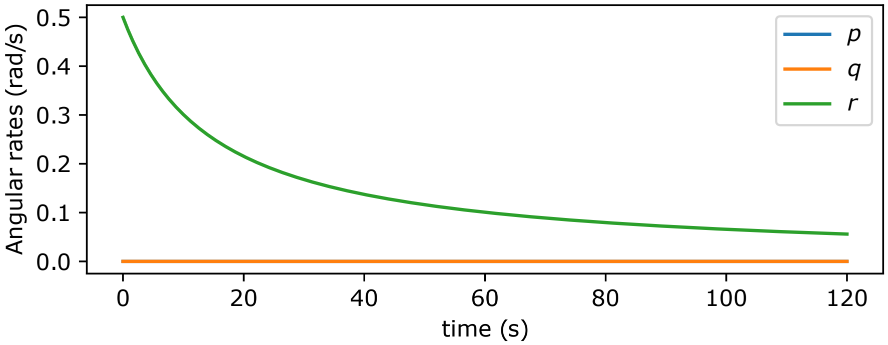


Figure 6.36. Yaw impulse response

6.5 CONCLUDING REMARKS

This chapter proposed a methodology for obtaining a meta-model for the hydrodynamic forces and moments that act on an ROV. The model is intended to be useful in time-domain simulations. The data used to compute regressions come from viscous-flow computations. In order to organise the computations, the language of design of computer experiments was used.

The process of obtaining a hydrodynamics meta-model was done for VISOR3, by using MARIN's viscous-flow solver ReFRESCO; the computations were arranged using three-circle designs. In all viscous-flow computations, the computational domain is a non-structured grid that uses a spherical outer boundary; the domain was computed using Hexpress.

Three types of viscous-flow computations were used: variations on incoming flow direction in steady state, circular manoeuvres in steady state, and imposed (unsteady) motion. The most intensive computations were done when incoming flow direction was varied. A future step should concentrate on the particularities of the other two types of computations.

The approach used in this thesis quantifies the numerical uncertainty due to different sources: domain size, level of grid refinement, iterative error, and that of simplifying the complete phenomenon by a single meta-model. The results indicate that the uncertainty due to simplifying by a meta-model that envelopes all the hydrodynamics is larger than that of other sources. Naturally, experimental validation are not included given the goals of this thesis; nevertheless, experimental validation is an important future step.

The foundations of performing computations using spherical space geometry, *i.e.*, spherical designs, were laid out in Chapters 4 and 5; nevertheless, spherical designs were not implemented in this chapter; as mentioned, the computations on VISOR3 were arranged using three-circle designs. From results in Sec. 5.4.6, three-circle designs are not a good option for regression purposes, because they deliver a larger integrated variance when the meta-models include cross-coupled terms, indicating that spherical arrangements are a better choice. Nonetheless, three-circle designs were considerably useful when interpreting the results in the main planes. This is because they allowed one to analyse the pertinence of using certain meta-model's basis functions, *i.e.*, based on how well they reproduce the observed geometry.

All told, an interesting future step could be to compute a meta-model of PIONERO500 and implement the aforementioned ideas that could not be fulfilled, such as make use of spherical designs to arrange the viscous-flow computations, study the particularities of other types of viscous-flow computations, and validate the results experimentally.

Chapter 7

Cable modelling

There are numerous approaches to modelling slender structures for underwater applications, such as catenary [64, 1, 66], beam/rod [62, 16, 19], collection of masses [35], or collection of rigid bodies [67, 113, 93]. Given that there is not previous knowledge about ROV cable modelling in UPB's ROV development group, this chapter may be considered as well as a local knowledge building block for further work. Through this work, the name *cable* is the generic way to refer to any slender structure related to an ROV system. Names such as catenary, beam, or rod are specific to the theory used to analyse cable mechanics.

A first step to study cable mechanics is to focus on the static and dynamic catenary formulation. This is done in order to test different modelling and numerical computation approaches, without taking into account the effect of cable's internal material mechanics; as a result, the usefulness of a catenary approach for ROV motion studies is tested. After that, linear elasticity is considered to study the effect of internal material mechanics.

This chapter focuses on solving steady-state and dynamic problems that are related and useful for ROV applications. Steady-state computations have two purposes: the first purpose is straight-forward, to obtain cable-induced loads in static (or pseudo-static) conditions; the second purpose is to properly initialise dynamic computations. Dynamic computations are the ultimate goal; they are useful to further predict the effect of the cable on ROV motion. Because cable models are non-linear, any numerical method requires proper initialisation. Consequently, the first problem consists in studying strategies to compute an initial geometry given operation conditions such as cable's attachment points, dimensions, weight, and current profile. The second problem consists in quantifying the amount of drag accumulation that the ROV withstands due to steady-state current profiles. Finally, the third problem consists in computing motion scenarios in dynamic conditions.

7.1 MOTION KINEMATICS IN CARTESIAN SPACE

Mechanically, cables are structural components where one of its dimensions is several orders of magnitude larger that the other (two) dimensions, *i.e.*, its length is several orders of magnitude larger than its diameter. Because of this, in a spatial sense, cables are one-dimensional structures and can be represented by a curve. Accordingly, the position of a cable point with respect to a NED frame $\{n\}$ is given by $\mathbf{r}(\mathbf{s}) \in \mathbb{R}^3$, where $s \in [0, L] \subset \mathbb{R}$ is the Lagrangian coordinate along its length. L is the total length and $\mathbf{r}(\mathbf{s})$ defines a curve that describes the cable geometry. See Fig. 7.1.

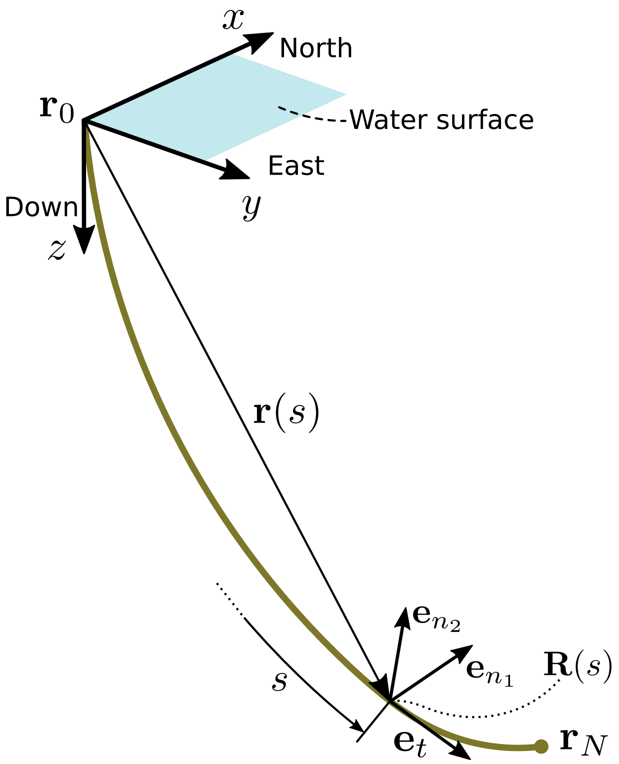


Figure 7.1. Cable description

In component form, $\mathbf{r}^n(s)$ can be written as

$$\mathbf{r}(\mathbf{s}) = \begin{bmatrix} x(s) & y(s) & z(s) \end{bmatrix}^{\mathsf{T}}. \tag{7.1}$$

As expected, x and y are North and East horizontal coordinates and z a vertical coordinate directed positive downwards; the vertical direction coincides with the direction of gravity. The tangential direction at any point is given by

$$\mathbf{e}_t(s) = \frac{\partial \mathbf{r}(\mathbf{s})}{\partial s}, \tag{7.2}$$

and in component form it may be written as

$$\frac{\partial \mathbf{r}}{\partial s} = \begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \end{bmatrix}^{\mathsf{T}}.$$

When loads can be placed on a plane, a two-dimensional kinematic formulation suffices. This is valid, for instance, where weight is the only external load. In such case $\mathbf{r}(\mathbf{s})$ can be written as

$$\mathbf{r}^n(s) = \begin{bmatrix} x(s) & z(s) \end{bmatrix}^{\mathsf{T}},$$

where x and z are the horizontal and vertical coordinates respectively. The tangential direction at any point is

$$\frac{\partial \mathbf{r}}{\partial s} = \begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial z}{\partial s} \end{bmatrix}^{\mathsf{T}}.$$

Orientation along the cable is represented by rotation matrix $\mathbf{R}(s)$ (see Fig. 7.1), which defines a frame continuum depending on s. There are many options for parametrising such matrix, and they often guarantee that one of the frame's basis vectors coincides with $\mathbf{e}_t$.

It is possible to parametrise $\mathbf{R}(s)$ by using Euler angles, this was proposed in [62]. Here, the successive rotations convention is the same as in [48]: first, a rotation around the $y-$axis an angle θ and, second, a rotation of the resulting frame around its $x-$axis an angle ϕ. When performing these rotations, it is assumed that $\mathbf{e}_t$ is initially aligned with the $z-$axis. More explicitly, it is possible to define

$$\mathbf{R}_c^n(s) = \mathbf{R}_c^n(\theta(s), \phi(s)) = \mathbf{R}_y(\theta)\mathbf{R}_x(\phi) = \begin{bmatrix} \cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}. \tag{7.3}$$

This matrix takes a vector on a cable's frame and obtains its representation on frame $\{n\}$. More explicitly, for an arbitrary vector in the cable's frame coordinates $\mathbf{V}^c = \begin{bmatrix} V_t & V_{n_1} & V_{n_2} \end{bmatrix}^{\mathsf{T}}$ one has

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \mathbf{R}_c^n(\theta, \phi) \begin{bmatrix} V_{n_1} \\ V_{n_2} \\ V_t \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} V_{n_1} \\ V_{n_2} \\ V_t \end{bmatrix} = \mathbf{R}_c^n(\theta, \phi)^{\mathsf{T}} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}.$$

From these results the cable frame's unitary basis vectors are given on $\{n\}$ as

$$\mathbf{e}^n_{n_1} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \end{bmatrix}^\mathsf{T},$$

$$\mathbf{e}^n_{n_2} = \begin{bmatrix} \sin\theta\sin\phi & \cos\phi & \cos\theta\sin\phi \end{bmatrix}^\mathsf{T}, \text{ and}$$

$$\mathbf{e}^n_{t} = \begin{bmatrix} \sin\theta\cos\phi & -\sin\phi & \cos\theta\cos\phi \end{bmatrix}^\mathsf{T}.$$

Additionally, according to (7.2), the components of $\mathbf{e}_t$ may be expressed as

$$\frac{\partial x}{\partial s} = \sin\theta\cos\phi, \tag{7.4}$$

$$\frac{\partial y}{\partial s} = -\sin\phi, \quad \text{and} \tag{7.5}$$

$$\frac{\partial z}{\partial s} = \cos\theta\cos\phi. \tag{7.6}$$

When variables are expressed on the cable's frame, derivation must account for magnitude variation as well as for direction variation. For instance, given an arbitrary vector $\mathbf{V}$ its total derivative with respect to space is given by the expression

$$\frac{\mathrm{D}\mathbf{V}}{\mathrm{D}s} = \frac{\partial \mathbf{V}}{\partial s} + \boldsymbol{\Omega} \times \mathbf{V}, \tag{7.7}$$

where $\boldsymbol{\Omega} = \begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 \end{bmatrix}^\mathsf{T}$, which accounts for direction variation. This vector, for transformation (7.3), is given by [48]

$$\boldsymbol{\Omega} = \begin{bmatrix} \frac{\partial\phi}{\partial s} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x(\phi)^\mathsf{T} \begin{bmatrix} 0 \\ \frac{\partial\theta}{\partial s} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial\phi}{\partial s} \\ \cos\phi\frac{\partial\theta}{\partial s} \\ -\sin\phi\frac{\partial\theta}{\partial s} \end{bmatrix}. \tag{7.8}$$

Additionally, the total derivative of $\mathbf{V}$ with respect to time is given by the expression

$$\frac{\mathrm{D}\mathbf{V}}{\mathrm{D}t} = \frac{\partial \mathbf{V}}{\partial t} + \omega \times \mathbf{V}, \tag{7.9}$$

where $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^\mathsf{T}$, which accounts for direction variation. This vector, for transformation (7.3), is given by [48]

$$\omega = \begin{bmatrix} \frac{\partial\phi}{\partial t} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x(\phi)^\mathsf{T} \begin{bmatrix} 0 \\ \frac{\partial\theta}{\partial t} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial\phi}{\partial t} \\ \cos\phi\frac{\partial\theta}{\partial t} \\ -\sin\phi\frac{\partial\theta}{\partial t} \end{bmatrix}. \tag{7.10}$$

7.2 EXTERNAL LOADS

In the models to come, the cable's load per unit length $\mathbf{q}$ is defined as

$$\mathbf{q} = \mathbf{w} + \mathbf{d}, \tag{7.11}$$

where $\mathbf{w}$ is weight and buoyancy and $\mathbf{d}$ is drag. More specifically $\mathbf{w}$ is assumed a constant field in Euclidean space and $\mathbf{d}$ a variable field. These fields are shown on a cable segment in Fig. 7.2 (a).

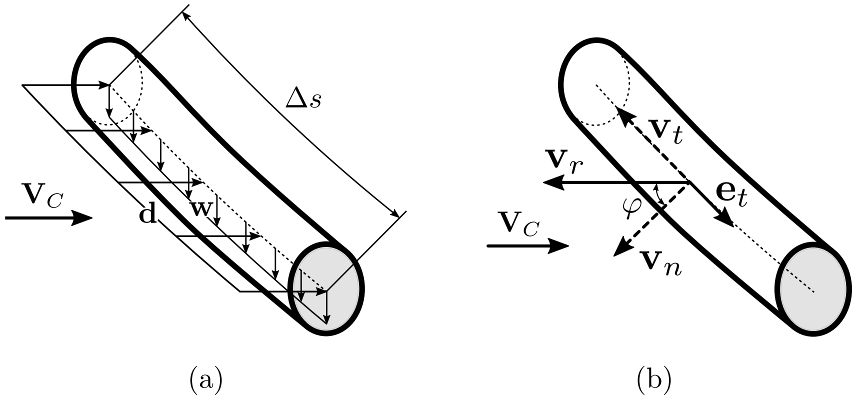


Figure 7.2. Cable loads

7.2.1 WEIGHT AND HYDROSTATICS

The load due to weight and buoyancy may be written in Cartesian coordinates with respect to $\{n\}$ as

$$\mathbf{w}^n = \begin{bmatrix} 0 & 0 & \bar{w} \end{bmatrix}^\intercal,$$

where $\bar{w}$ is the cable's weight in water per unit length, defined as $\bar{w} = \rho_c A g - \rho_\mathrm{w} A g$. In this expression, ρ_c is cable density, ρ_w water density, g acceleration due to gravity, $A = \frac{\pi}{4} d^2$ cross-section area, and d diameter.

Load $\mathbf{w}^n$ is directed along the z-axis of $\{n\}$. In the cable frame and using (7.3), it is given as

$$\mathbf{w}^c = \mathbf{R}_c^n(\theta, \phi)^\intercal \begin{bmatrix} 0 \\ 0 \\ \bar{w} \end{bmatrix} = \begin{bmatrix} -\bar{w}\sin\theta \\ \bar{w}\cos\theta\sin\phi \\ \bar{w}\cos\theta\cos\phi \end{bmatrix}. \tag{7.12}$$

7.2.2 DRAG DUE TO CURRENT

Drag is dependent on the cable's velocity relative to water. Relative velocity is defined as $\mathbf{v}_r = \mathbf{v} - \mathbf{V}_C$, where $\mathbf{v} = \mathbf{v}(s)$ is the velocity of a cable element and $\mathbf{V}_C$ is current. For irrotational currents, angular velocities are ignored. In inertial frame $\{n\}$, $\mathbf{V}_C^n$ is given by $\mathbf{V}_C^n = \begin{bmatrix} V_{C_x} & V_{C_y} & V_{C_z} \end{bmatrix}^\mathsf{T}$.

As shown in Fig. 7.2 (b), relative velocity $\mathbf{v}_r$ can be written in tangential and normal components, *i.e.*, $\mathbf{v}_r = \mathbf{v}_t + \mathbf{v}_n$. Tangential velocity is defined as $\mathbf{v}_t = (\mathbf{v}_r^\mathsf{T}\mathbf{e}_t)\,\mathbf{e}_t$ and normal velocity as $\mathbf{v}_n = \mathbf{v}_r - \mathbf{v}_t$. This distinction between tangential and normal velocities acknowledges that drag effects are different over those two directions. Accordingly, drag per unit length is defined as

$$\mathbf{d} = -\frac{1}{2}\rho_\mathrm{w} d\big(C_t|\mathbf{v}_t|\mathbf{v}_t + C_n|\mathbf{v}_n|\mathbf{v}_n\big), \tag{7.13}$$

where ρ_w is water density, d cable diameter, and C_t and C_n drag coefficients. In [46], such coefficients are defined as $C_t = C_d\mathrm{f}_t(\varphi)$ and $C_n = C_d\mathrm{f}_n(\varphi)$, where f_t and f_n are load coefficient functions dependent on incidence angle φ and C_d is drag coefficient.

Using the cable frame, relative velocity is given as $\mathbf{v}_r^c = \begin{bmatrix} v_{n_1} & v_{n_2} & v_t \end{bmatrix}^\mathsf{T}$ and velocity as $\mathbf{v}^c = \begin{bmatrix} v_{n_1}^c & v_{n_2}^c & v_t^c \end{bmatrix}^\mathsf{T}$. Using (7.3), relative velocity becomes

$$\begin{aligned}\begin{bmatrix} v_{n_1} \\ v_{n_2} \\ v_t \end{bmatrix} &= \begin{bmatrix} v_{n_1}^c \\ v_{n_2}^c \\ v_t^c \end{bmatrix} - \mathbf{R}_c^n(\theta,\phi)^\mathsf{T}\begin{bmatrix} V_{C_x} \\ V_{C_y} \\ V_{C_z} \end{bmatrix} \\ &= \begin{bmatrix} v_{n_1}^c - V_{C_x}\cos\theta - V_{C_z}\sin\theta \\ v_{n_2}^c - V_{C_x}\sin\theta\sin\phi + V_{C_y}\cos\phi + V_{C_z}\cos\theta\sin\phi \\ v_t^c - V_{C_x}\sin\theta\cos\phi + V_{C_y}\sin\phi + V_{C_z}\cos\theta\cos\phi \end{bmatrix}. \end{aligned} \tag{7.14}$$

Thus, the drag vector for unit length is given by

$$\mathbf{d} = -\frac{1}{2}\rho_\mathrm{w} dC_d \begin{bmatrix} f_n(\varphi)v_{n_1}\sqrt{v_{n_1}^2 + v_{n_2}^2} \\ f_n(\varphi)v_{n_2}\sqrt{v_{n_1}^2 + v_{n_2}^2} \\ f_t(\varphi)v_t|v_t| \end{bmatrix}. \tag{7.15}$$

7.2.3 ADDED MASS

Hydrodynamic *added* mass depends on relative acceleration and is often considered as an increase of the mass. As a result, relative acceleration is defined as $\mathbf{a}_r = \mathbf{a} - \mathbf{a}_C$, where $\mathbf{a} = \mathbf{a}(s)$ is cable acceleration and, furthermore, $\mathbf{a}_C = \mathbf{0}$. Hence, $\mathbf{a}_r = \mathbf{a}$.

Similarly as with relative velocity, acceleration $\mathbf{a}$ can be written in tangential and normal components, *i.e.*, $\mathbf{a} = \mathbf{a}_t + \mathbf{a}_n$. Tangential acceleration is defined as $\mathbf{a}_t = (\mathbf{a}^\intercal \mathbf{e}_t)\, \mathbf{e}_t$ and normal acceleration as $\mathbf{a}_n = \mathbf{a} - \mathbf{a}_t$. This distinction between tangential and normal acceleration allows one to assume that added mass is a normal effect only. Accordingly, added mass load is

$$\mathbf{m}_a = -m_a \mathbf{a}_n. \tag{7.16}$$

Subsequently, a complete inertial term becomes $m\mathbf{a} + m_a \mathbf{a}_n$, where m_a accounts for added mass per unit length and $m = \rho_c A$ for mass per unit length. Throughout this work, it is assumed that added mass is equivalent to the mass of the displaced volume, *i.e.*, $m_a = \rho_\mathrm{w} A$.

Using the cable frame, mass and added mass can be expressed by 3×3 matrix

$$\mathbf{m}^c = \mathrm{diag}\{m + m_a, m + m_a, m\}.$$

So that the inertial term becomes $\mathbf{m}^c \mathbf{a}^c$.

7.3 INEXTENSIBLE CATENARY STATICS IN CARTESIAN SPACE

A cable may be considered an *inextensible catenary* if it does not accumulate energy through deformation. More specifically, this means that it does not deform axially and bends and twists without resistance. The inextensible catenary is the most basic representation of a cable and a starting point for constructing the required building blocks to study ROV cable mechanics. This section considers the classic catenary in steady-state, subject to a constant field per unit length. The solutions provided here are mainly useful to compute initial conditions for ROV cable problems in steady-state and dynamic conditions.

The formulation herein uses catenary theory available in [64, 120]. The main assumption is that the cable only sustains tensile forces. To frame a model, suppose that an inextensible catenary hangs between two fixed points, namely $\mathbf{r}_0$ and $\mathbf{r}_N$. Then, for a static inextensible catenary, the forces balance is given by

$$\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}s} + \mathbf{q} = \mathbf{0}, \tag{7.17}$$

where $\mathbf{F}$ is the internal force vector and $\mathbf{q}$ is the external forces field vector per unit length, measured with respect an inertial frame. One assumes that there are no moment loads and, as mentioned, $\mathbf{F}$ is always tangent to the catenary curve. This may be expressed as

$$\mathbf{F} = ||\mathbf{F}||\mathbf{e}_t = ||\mathbf{F}||\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}s}, \tag{7.18}$$

where $||\mathbf{F}||$ is the magnitude of $\mathbf{F}$ that describes the catenary's internal tension T, *i.e.*, $T = ||\mathbf{F}||$. Please note that derivatives are not partial because s is the only independent variable in this problem.

This problem can be reduced to two-dimensions. In this case $\mathbf{F}$ is given by

$$\mathbf{F} = \begin{bmatrix} F_x \\ F_z \end{bmatrix}$$

and the magnitude by $||\mathbf{F}|| = \sqrt{F_x^2 + F_z^2}$. When weight is the only external load, the force field may be written in two-dimensional Cartesian coordinates as

$$\mathbf{q} = \mathbf{w} = \begin{bmatrix} 0 \\ \bar{w} \end{bmatrix},$$

where w is the catenary's weight in water per unit length.

7.3.1 ANALYTIC SOLUTION

Provided that position and conditions at the left boundary, namely at $s = 0$, are known and weight is the only load, Eq. (7.17) has analytical solution. Such boundary conditions may be written as $\mathbf{r}_0 = \mathbf{r}(0) = \begin{bmatrix} x_0 & z_0 \end{bmatrix}^\mathsf{T}$ and $\mathbf{F}(0) = \begin{bmatrix} H & V \end{bmatrix}^\mathsf{T}$, with $V > 0$. Conversely, (7.17) becomes

$$\frac{\mathrm{d}}{\mathrm{d}s} \begin{bmatrix} F_x \\ F_z \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{w} \end{bmatrix} = \mathbf{0}, \tag{7.19}$$

If one integrates (7.19) from $s = 0$ to an arbitrary length s, it becomes

$$\mathbf{F} = \begin{bmatrix} F_x \\ F_z \end{bmatrix} = \begin{bmatrix} H \\ V - \bar{w}s \end{bmatrix}. \tag{7.20}$$

According to (7.20), the horizontal load is constant through all the length; the vertical load varies linearly.

Then, considering that $||\mathbf{F}|| = \sqrt{F_x^2 + F_z^2} = \sqrt{H^2 + (V - \bar{w}s)^2}$, an expression for the cable geometry can be established. Using the definitions in (7.1), (7.2), (7.18) and (7.20) one obtains the expression

$$\begin{bmatrix} H \\ V - \bar{w}s \end{bmatrix} = \sqrt{H^2 + (V - \bar{w}s)^2} \begin{bmatrix} \frac{\mathrm{d}x}{\mathrm{d}s} \\ \frac{\mathrm{d}z}{\mathrm{d}s} \end{bmatrix},$$

which arranged and in integral form is

$$\int_{x_0}^{x} \mathrm{dx} = \int_0^s \frac{H}{\sqrt{H^2 + (V - \bar{w}\mathrm{s})^2}} \mathrm{ds},$$
$$\int_{z_0}^{z} \mathrm{dz} = \int_0^s \frac{V - \bar{w}\mathrm{s}}{\sqrt{H^2 + (V - \bar{w}\mathrm{s})^2}} \mathrm{ds}.$$

The integral gives an expression for x and z coordinates, namely

$$\begin{aligned} x =& x_0 + \frac{|H|}{\bar{w}}\left(\operatorname{asinh}\left(\frac{V}{H}\right) - \operatorname{asinh}\left(\frac{V-\bar{w}s}{H}\right)\right), \\ z =& z_0 + \frac{1}{\bar{w}}\left(\sqrt{H^2+V^2} - \sqrt{H^2+(V-\bar{w}s)^2}\right). \end{aligned}$$

Using these expressions, provided that left-boundary force is known, it is possible to calculate the catenary's coordinates at any given length s. For instance, the right boundary coordinates x_L and z_L, when $s = L$ is the total length, are

$$\begin{aligned} x_L = x(L) =& x_0 + \frac{|H|}{\bar{w}}\left(\operatorname{asinh}\left(\frac{V}{H}\right) - \operatorname{asinh}\left(\frac{V-\bar{w}L}{H}\right)\right), \\ z_L = z(L) =& z_0 + \frac{1}{\bar{w}}\left(\sqrt{H^2+V^2} - \sqrt{H^2+(V-\bar{w}L)^2}\right). \end{aligned}$$

7.3.2 CANONICAL CATENARY SOLUTION

The catenary may be represented using non-dimensional quantities. To do so, all distances are made non-dimensional using the total cable length L and all loads are made non-dimensional using the total cable weight in water $\bar{w}L$; moreover, it is assumed that $x_0 = z_0 = 0$ without losing generality. Consequently, non-dimensional distances and forces become $x_\delta = x/L$, $z_\delta = z/L$, $s_\delta = s/L$, $h = H/(\bar{w}L)$, $v = V/(\bar{w}L)$, $f_x = F_x/(\bar{w}L)$, and $f_z = F_z/(\bar{w}L)$. This gives the expressions

$$\begin{aligned} f_x =& h \quad \text{and} \\ f_z =& v - s_\delta \end{aligned}$$

for the forces and

$$\begin{aligned} x_\delta &= |h|\left(\operatorname{asinh}\left(\frac{v}{h}\right) - \operatorname{asinh}\left(\frac{v-s_\delta}{h}\right)\right) \quad \text{and} \\ z_\delta &= \sqrt{h^2+v^2} - \sqrt{h^2+(v-s_\delta)^2} \end{aligned}$$

for the coordinates. Additionally, the expression of the end coordinates becomes

$$x_{\delta_N} = x_\delta(1) = |h|\left(\operatorname{asinh}\left(\frac{v}{h}\right) - \operatorname{asinh}\left(\frac{v-1}{h}\right)\right) \quad \text{and} \tag{7.21a}$$

$$z_{\delta_N} = z_\delta(1) = \sqrt{h^2+v^2} - \sqrt{h^2+(v-1)^2}. \tag{7.21b}$$

7.3.3 SHOOTING ALGORITHM

As formulated, the catenary solution allows one to calculate the forces and coordinates when the left boundary conditions are known. This problem corresponds to an initial-value problem. If conversely one wants to know the forces and geometry of the catenary when the right and left boundary coordinates are specified, the proposed solution is not suited to do so. This problem corresponds to a boundary-valued problem. An alternative to solving this problem consists in computing the so-called shooting algorithm, where code iterates over the initial conditions, or left boundary conditions, and solves the initial-value problem until the right boundary conditions are met.

Because the initial-value problem has analytical solution, the shooting algorithm may be solved as a non-linear algebraic equation. This is equivalent to finding the roots of the equation

$$\mathrm{f}(h,v) = \begin{bmatrix} \mathrm{f}_1(h,v) \\ \mathrm{f}_2(h,v) \end{bmatrix} = \begin{bmatrix} |h| \left(\operatorname{asinh}\left(\frac{v}{h}\right) - \operatorname{asinh}\left(\frac{v-1}{h}\right)\right) - x_{\delta_N} \\ \sqrt{h^2+v^2} - \sqrt{h^2+(v-1)^2} - z_{\delta_N} \end{bmatrix}, \tag{7.22}$$

given any combination of x_{δ_N} and z_{δ_N}.

The Jacobian of $\mathrm{f}(h,v)$, which is used for instance in a root-finding algorithm, is given by

$$\mathbf{J}_f(h,v) = \begin{bmatrix} \frac{\partial \mathrm{f}_1}{\partial h} & \frac{\partial \mathrm{f}_1}{\partial v} \\ \frac{\partial \mathrm{f}_2}{\partial h} & \frac{\partial \mathrm{f}_2}{\partial v} \end{bmatrix}, \tag{7.23}$$

where

$$\begin{aligned}
\frac{\partial \mathrm{f}_1}{\partial h} =& \operatorname{sign}(h)\left(\operatorname{asinh}\left(\frac{v}{h}\right) - \operatorname{asinh}\left(\frac{v-1}{h}\right)\right) \\
&+ \frac{v-1}{\sqrt{h^2+(v-1)^2}} - \frac{v}{\sqrt{h^2+v^2}}, \\
\frac{\partial \mathrm{f}_1}{\partial v} =& \frac{|h|}{\sqrt{h^2+v^2}} - \frac{|h|}{\sqrt{h^2+(v-1)^2}}, \\
\frac{\partial \mathrm{f}_2}{\partial h} =& \frac{h}{\sqrt{h^2+v^2}} - \frac{h}{\sqrt{h^2+(v-1)^2}}, \quad \text{and} \\
\frac{\partial \mathrm{f}_2}{\partial v} =& \frac{v}{\sqrt{h^2+v^2}} - \frac{v-1}{\sqrt{h^2+(v-1)^2}}.
\end{aligned}$$

7.3.4 NUMERICAL RESULTS ON THE ANALYTIC SOLUTION

As indicated by (7.22), for any given x_{δ_N} and z_{δ_N} there are h and v that satisfy the equation. Two things can be stated: first, as suggested previously, x_{δ_N} and z_{δ_N} can be found by using a root-finder algorithm over (7.22); and second, there are functions h and v so that $h = \mathrm{h}(x_{\delta_N}, z_{\delta_N})$ and $v = \mathrm{h}(x_{\delta_N}, z_{\delta_N})$ represent the same set as (7.22). For the former, if a root-finder algorithm is to be used, initial conditions are needed, and for the latter, given the absence of analytical solution, they may be approximated by using numerical solutions and, for instance, a polynomial regression model.

A combination of the two previously mentioned elements may be used as follows: use the results of numerical computations to find meta-models for h and v, and use the results of the meta-model to initialise a shooting algorithm. This has the advantage that the shooting algorithm is initialised near the solution, hence requiring less iterations. This will be explored later.

All root-finding problems of the form $\mathrm{f}(\mathbf{x}) = \mathbf{0}$, where $\mathtt{x}$ is a vector of unknowns, are solved by using Python's ScyPy optimisation module. The `fsolve` function is used, which is a wrapper that uses FORTRAN-MINPACK's `hybrd` and `hybrj` algorithms [111]. All default options work well on all problems, unless explicitly stated. In all the cases the Jacobian of the function is explicitly declared to the solver, for instance, (7.23) is provided along (7.22).

7.3.5 PROBLEM'S DOMAIN

Before performing further computations, the problem's domain is parametrised. The target position is given by $x_{\delta_N} = r_l \sin\theta_l$ and $z_{\delta_N} = r_l \cos\theta_l$. The problem is solved only in the first quadrant, this means that $\theta \in [0, \pi/2]$. The solutions in the second quadrant are symmetric to the ones of the first quadrant (the functions that define x_{δ_N} and z_{δ_N} are even). The limits of r_l, given that the catenary is inextensible, are strictly between 0 and 1. Nevertheless, values approaching 1 require the catenary to be highly tensioned. Such results are not in the interests of this work. For this reason a different range for r_l is to be used. In this work, values for r_l such that $\sqrt{h^2 + v^2} < 1$ are used. This means that tension at the left boundary is less than total weight.

The problem's boundary may be found by setting $h = \sin\theta_f$ and $v = \cos\theta_f$ in (7.21), enforcing unitary tension at the left boundary. This gives the expressions

$$x_{\delta_N \max} = |\sin\theta_f| \left(\operatorname{asinh}\left(\frac{\cos\theta_f}{\sin\theta_f}\right) - \operatorname{asinh}\left(\frac{\cos\theta_f - 1}{\sin\theta_f}\right)\right) \quad \text{and} \tag{7.24a}$$

$$z_{\delta_N \max} = 1 - \sqrt{\sin{\theta_f}^2 + (\cos\theta_f - 1)^2}. \tag{7.24b}$$

The limits of the problem are given by the set defined in (7.24a) and (7.24b), and the results are illustrated in Fig. 7.3.

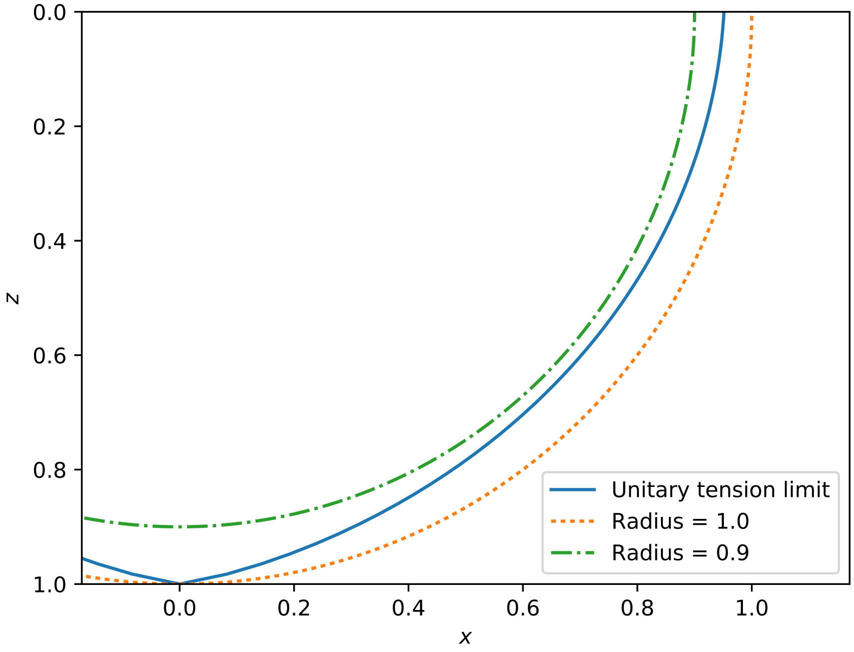


Figure 7.3. Problem domain

7.3.6 ENERGY FUNCTION

The shooting algorithm can be understood, according to Eq. (7.22), as finding the roots of f. Conversely, they can be understood as well as a minimisation problem of the functional

$$J(h, v) = \frac{1}{2}\,\mathrm{f}^{\intercal}(h, v)\,\mathrm{f}(h, v). \tag{7.25}$$

This can be seen as a potential energy functional related to the problem. Moreover, the gradient of such function can be computed as

$$\nabla J(h, v) = \mathrm{f}^{\intercal}(h, v)\mathbf{J}_f(h, v), \tag{7.26}$$

where $\mathbf{J}_f(h, v)$ is calculated according to (7.23).

7.3.7 ILLUSTRATIVE CASES

For the sake of illustration, different solutions of the problem are computed. The solutions presented herein contain all the possible combinations for θ_l = 15, 45 and 75 deg and r_l = 0.25, 0.50 and 0.75. Each of the cases has the following computations: 1) computation of the shooting algorithm to find h and v such the catenary reaches the required right boundary position, 2) computation of the resultant catenary geometry, *i.e.*, x_δ and z_δ coordinates, and 3) evaluation of the energy functional (7.25).

The results of the computations are shown in Figs. 7.4 and 7.5. Figure 7.4 shows the resultant geometries. Moreover, Fig. 7.5 illustrates the contours of the energy functions: each row represents a different value of θ_l, increasing from up to down. Each column represents a different value of r_l, increasing from left to right. Additionally, the dot represents the solution to the shooting algorithm problem.

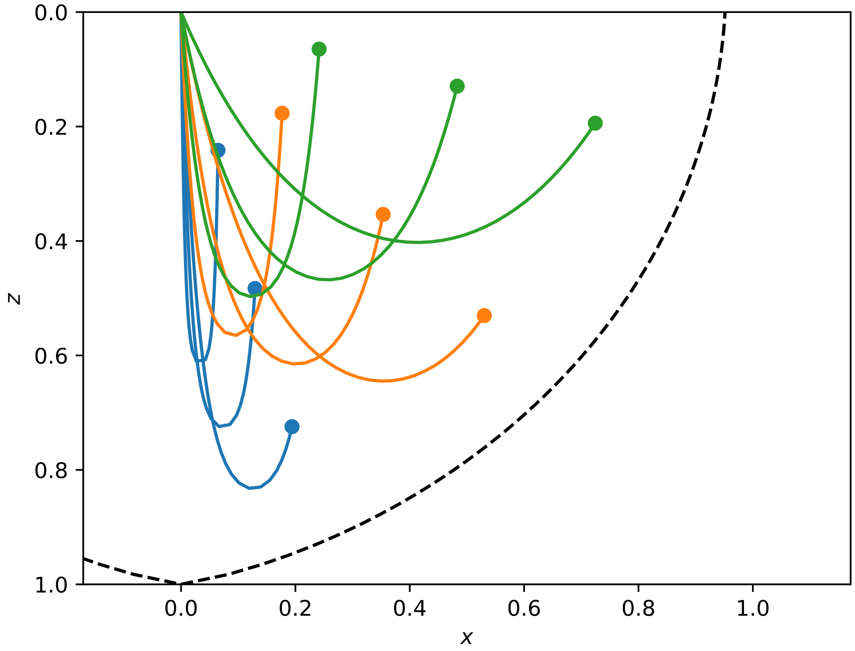


Figure 7.4. Catenary resultant geometry

7.3.8 INITIAL CONDITIONS

The objective of this section is to find suitable initial conditions for the shooting algorithm. To do such thing, several computations are performed in a Monte-Carlo fashion. For each iteration pseudo-random values are chosen for r_l and θ_l inside the domain of the problem. For these conditions, the shooting algorithm is computed to find the corresponding values of h and v. Each Monte-Carlo run consist of N iterations; in this case $N = 1 \times 10^6$ is chosen. Three different Monte-Carlo runs are performed like this:

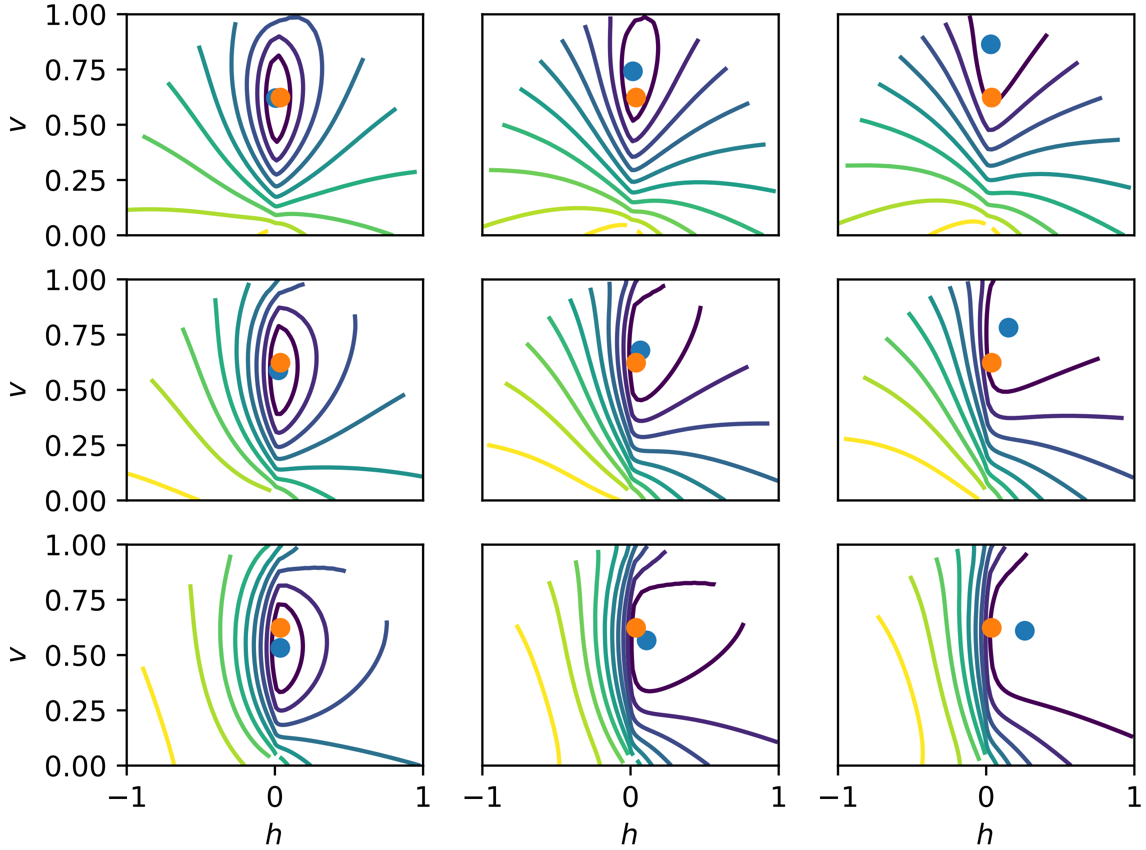


Figure 7.5. Energy function contours

1. Ad-hoc values of h and v are chosen to initialise the root-finding algorithm. In this case, values of $h = 0.2$ and $v = 0.5$ are used.
2. From the previous results, calculate the 50 % percentile of all computed values of h and v and use these values to initialise the root finding algorithm.
3. From the previous results, find a regression model that allows the estimation of values of h and v as a function of r_l and/or θ_l to better-initialise the root-finding algorithm. To find a suitable regression model, a covariance analysis is performed to select the variables to be used in a regression polynomial.

To check the effectiveness of each initial-condition option, the number of evaluations of the function performed by the root-finding algorithm at each iteration are computed and compared.

The results overview is as follows:

1. With the chosen *ad-hoc* values of h and v the iterations' number of function evaluations have a 50 % percentile of 10. Excluding outliers, the number of evaluations range between 9 and 12.
2. The 50 % percentile values of h and v, rounded to the fourth decimal, are $h = 0.0375$ and $v = 0.6233$. Using these values as initial conditions the iterations' number of function evaluations have a 50 % percentile of 8 and, excluding outliers, the number of evaluations range between 4 and 12. These initial conditions represent a better option compared to previous *ad-hoc* values.
3. From the previous results, the resultant regression model is given by

   $$\begin{aligned} h_{\text{init}} =& 2.87784412x_{\delta_N}^5 - 4.49604799x_\delta^4 + 2.73869239x_\delta^3 - 0.381152054x_\delta^2 \\ &+ 0.149027798x_\delta, \end{aligned} \tag{7.27}$$

   $$v_{\text{init}} = -0.10729293z_\delta^3 + 0.09862232z_\delta^2 + 0.50221887z_\delta + 0.5. \tag{7.28}$$

   The derivation of these functions will be further discussed. Using these functions to compute the initial conditions the iterations' number of function evaluations have a 50 % percentile of 5 and, excluding outliers, the number of evaluations range between 4 and 7. This one, compared to the previous two, represent the best option for initialising the root-finding algorithm.

The results discussed herein are summarised in Fig. 7.6. To obtain the regression model two

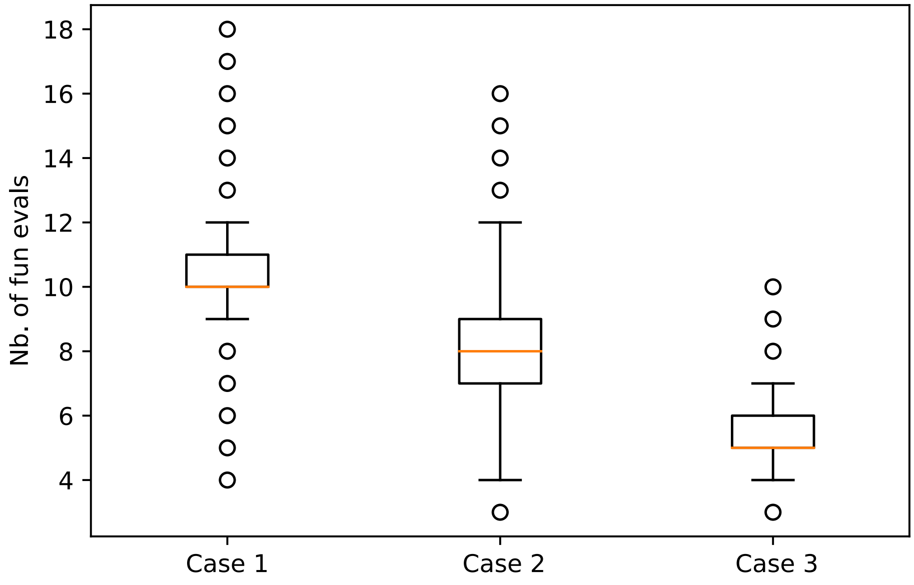


Figure 7.6. Number of function evaluations at each run

steps are followed: first, a covariance analysis is used to choose suitable model's independent variables; second, different polynomial regressions are tested and, based on the behaviour of the residuals, a polynomial degree is chosen.

The correlation coefficients of variables h, v, $\sqrt{h^2+v^2}$, atan(h/v), x_{δ_N}, z_{δ_N}, r_l, and θ_l were computed; however, h, v, x_{δ_N}, and z_{δ_N} where considered enough to construct a regression model. Figure 7.7 shows scatter plots of h and v as functions of x_{δ_N} and z_{δ_N} resulting from all iterations. For illustration purposes, only 0.2 % of the total computed data is shown in the figure. Additionally, the shade of grey represents the amount of correlation: white and black correspond to correlation coefficients of 0 and 1 respectively.

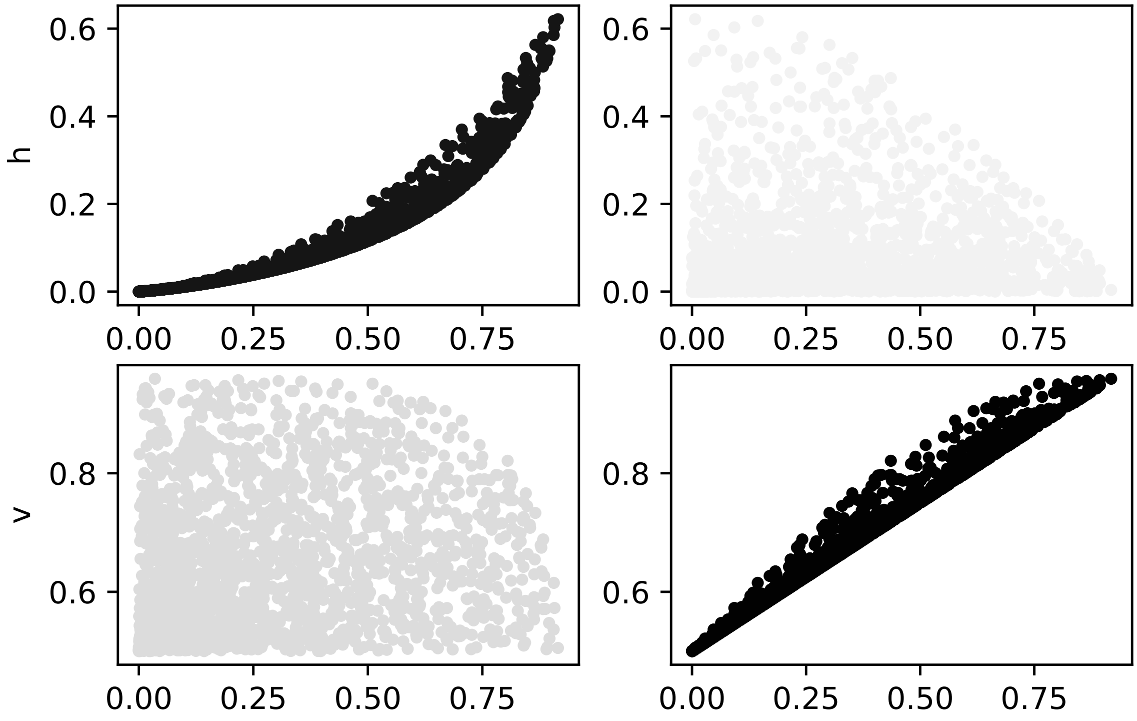


Figure 7.7. Correlation plots

Figure 7.8 shows the residual values resultant from choosing polynomials of different degrees as regression functions, as well as the difference between consecutive residuals. The degree of the polynomial used for the approximated function is chosen so that the difference of consecutive residuals reaches at least 1. This gives a polynomials of degrees 5 and 3 for h and v respectively.

Tables 7.1 and 7.2 show all polynomials' coefficients from degrees 0 to 5. Coefficients' are truncated to the fourth decimal and residuals are presented as whole numbers.

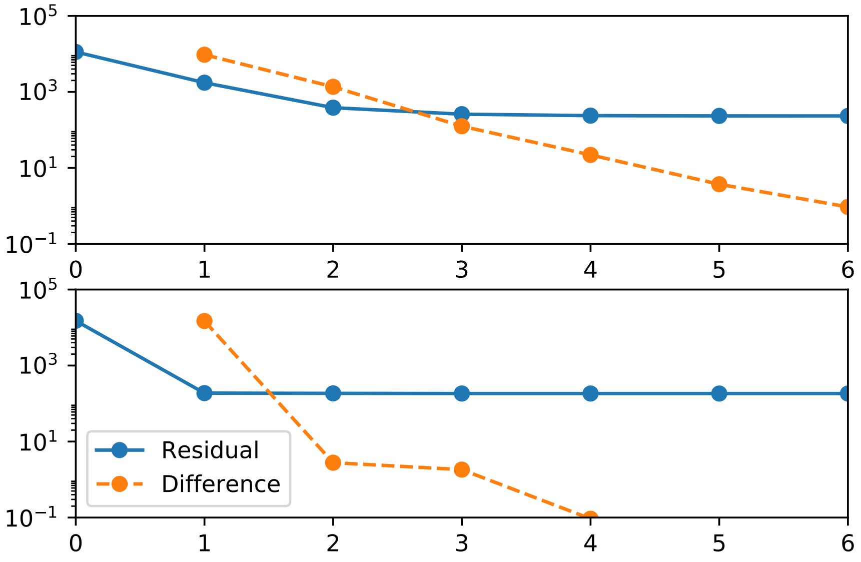


Figure 7.8. Residuals plots

Table 7.1. Regression polynomials coefficients for h

Degree	1	x	x^2	x^3	x^4	x^5	Residual
0	0.0806						11268
1	−0.0409	0.4147					1747
2	0.0117	−0.0906	0.6762				384
3	−0.006	0.2338	−0.3923	0.8867			259
4	0.0021	−0.0044	1.0027	−1.7959	1.6126		237
5	−0.001	0.1490	−0.3812	2.7387	−4.4960	2.8778	234

The zero crossings of the functions may be found analytically from (7.21). When $x_{\delta_N} = 0$, $h = 0$, and for $z_{\delta_N} = 0$, $v = 1/2$. Moreover, if $h = 0$ and $0 < v < 1$ are assumed, then $v = \frac{1+z_{\delta_N}}{2}$. This can be considered while further constructing regression functions. The regression functions (7.28) will be used to give an initial guess to the shooting algorithm. These functions, along with scatter data of 0.2 % of the simulations, are shown in Fig. 7.9.

As a practical recommendation from this section, if a unique set of values for h and v is to be used then initialising the algorithm with $h = 0.0375$ and $v = 0.6233$ is a good choice. A better choice is to use functions (7.28) and compute initial values depending on the right boundary coordinates.

Table 7.2. Regression polynomials coefficients for v

Degree	1	x	x^2	x^3	x^4	x^5	Residual
0	0.6523						15052
1	0.5006	0.5186					188
2	0.4982	0.5414	−0.03058				185
3	0.5003	0.5022	0.0986	−0.1073			183
4	0.4998	0.5177	0.0079	0.0672	−0.1049		183
5	0.5000	0.5101	0.0762	−0.1567	0.1967	−0.1421	183

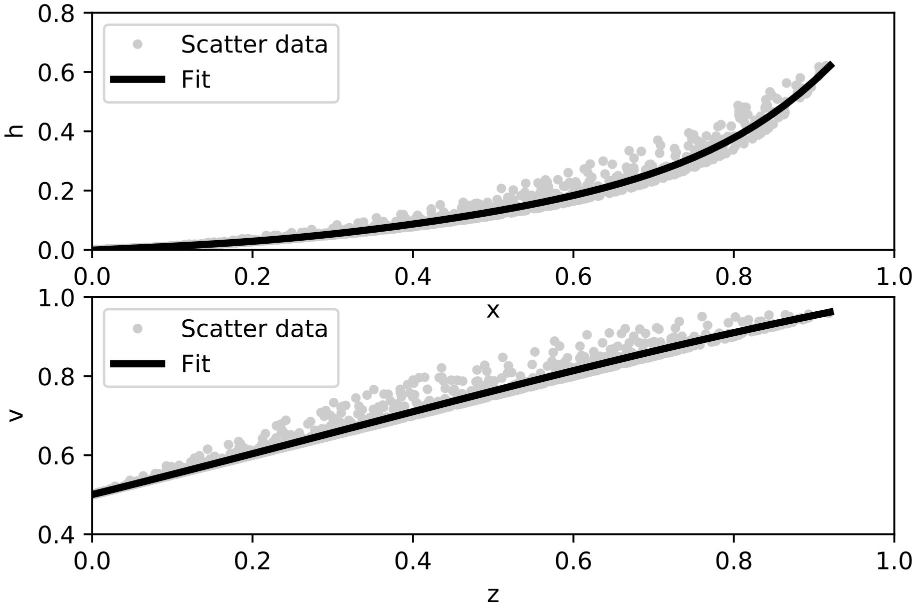


Figure 7.9. Regression functions

7.3.9 OPTIMAL CATENARY CONFIGURATION

For the catenary, an optimisation problem can be stated as follows: if a position in the xz-plane should be reached at the right boundary, find the optimal cable length so that the tension at the left boundary is a minimum. This may be stated as

$$\min \frac{1}{2}||T_{\delta_N}||^2,$$

subject to the catenary model.

For this problem, a different model description is formulated in order to include the cable length as an optimisation variable. More conceptually, it is necessary to include the fact that if the cable length is greater then the cable tension should be greater as well; in opposition, if the catenary length approaches the span between the two boundaries, the tension increments also. These opposed effects should derive an optimal configuration placed in between.

This new formulation requires a non-dimensional model different to the canonical one. Let the coordinates of the left boundary be given by $x_N = r_l \sin\theta_l$ $z_N = r_l \cos\theta_l$. Then, r_l is used to make all distances non-dimensional and wr_l to make all loads non-dimensional. Consequently, one defines that $x_\delta = x/r_l$, $z_\delta = z/r_l$, $s_\delta = s/r_l$, $l = L/r_l$, $h = H/(wr_l)$, $x_{\delta_N} = \sin\theta_l$ $z_{\delta_N} = \cos\theta_l$, $v = V/(wr_l)$, $f_x = F_x/(wr_l)$, and $f_z = F_z/(wr_l)$.

Using these definitions, the forces and coordinates model look similar to the canonical catenary's but gives a different description of the right boundary loads and coordinates. The right boundary load is given by

$$\begin{aligned} f_{x_N} =& h \quad \text{and} \\ f_{z_N} =& v - l \end{aligned}$$

and the right boundary coordinates are given by

$$x_{\delta_N} = x_\delta(1) = |h| \left(\operatorname{asinh}\left(\frac{v}{h}\right) - \operatorname{asinh}\left(\frac{v-l}{h}\right)\right) \quad \text{and} \tag{7.29a}$$

$$z_{\delta_N} = z_\delta(1) = \sqrt{h^2+v^2} - \sqrt{h^2+(v-l)^2}. \tag{7.29b}$$

If the right boundary load is to be minimised, then the target function may be defined as

$$f_T(h,v,l) = \frac{1}{2}\left(h^2 + (v-l)^2\right)$$

and its gradient by

$$\nabla f_T = \begin{bmatrix} h, & v-l, & l-v \end{bmatrix}^{\mathsf{T}}$$

The catenary model (7.29a) and (7.29b) generates the constraints

$$\begin{aligned} c_1(h,v,l) =& |h| \left(\operatorname{arcsinh}\left(\frac{v}{h}\right) - \operatorname{arcsinh}\left(\frac{v-l}{h}\right)\right) - \sin\theta_l, \\ c_2(h,v,l) =& \sqrt{h^2+v^2} - \sqrt{h^2+(v-l)^2} - \cos\theta_l. \end{aligned}$$

The gradients of these functions are

$$\nabla c_1(h,v,l) = \begin{bmatrix} \operatorname{sign}(h)\left(\operatorname{asinh}\left(\frac{v}{h}\right) - \operatorname{asinh}\left(\frac{v-l}{h}\right)\right) + \frac{v-1}{\sqrt{h^2+(v-l)^2}} - \frac{v}{\sqrt{h^2+v^2}} \\ \frac{|h|}{\sqrt{h^2+v^2}} - \frac{|h|}{\sqrt{h^2+(v-l)^2}} \\ \frac{|h|}{\sqrt{h^2+(v-l)^2}} \end{bmatrix}$$

and

$$\nabla c_2(h,v,l) = \begin{bmatrix} \frac{h}{\sqrt{h^2+v^2}} - \frac{h}{\sqrt{h^2+(v-l)^2}} \\ \frac{v}{\sqrt{h^2+v^2}} - \frac{v-l}{\sqrt{h^2+(v-l)^2}} \\ \frac{v-l}{\sqrt{h^2+(v-l)^2}} \end{bmatrix}.$$

The solution to this problem is illustrated in Fig. 7.10. A practical solution to this problem is shown in Fig. 7.11. Here, the catenary length to span l/r_l is plotted as a function of θ_l, namely the direction angle of the right boundary coordinate. The optimal ratio is compared to the minimum ratio given by the boundaries in (7.24a) and (7.24b): ratios below the minimum herein will represent a non-dimensional tension over 1.

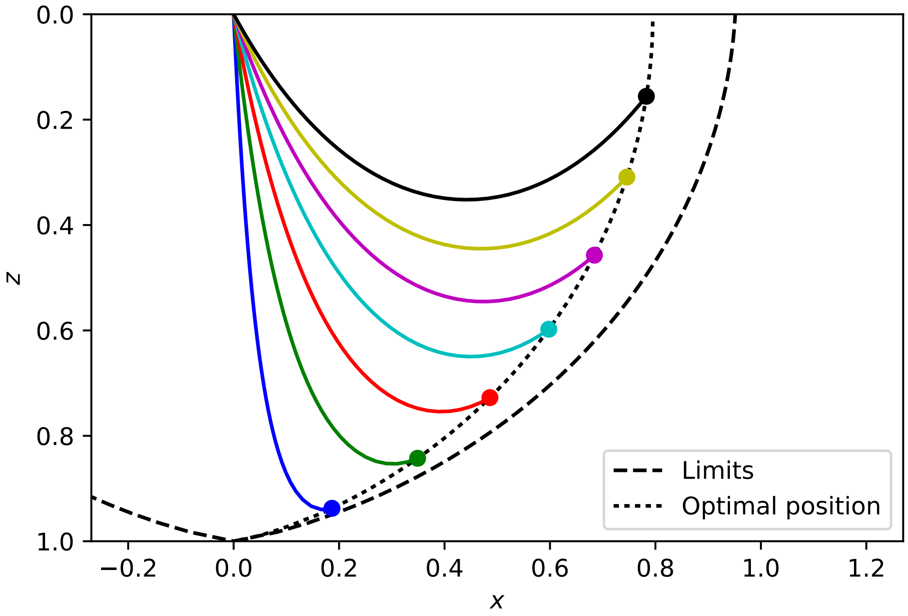


Figure 7.10. Optimal configuration

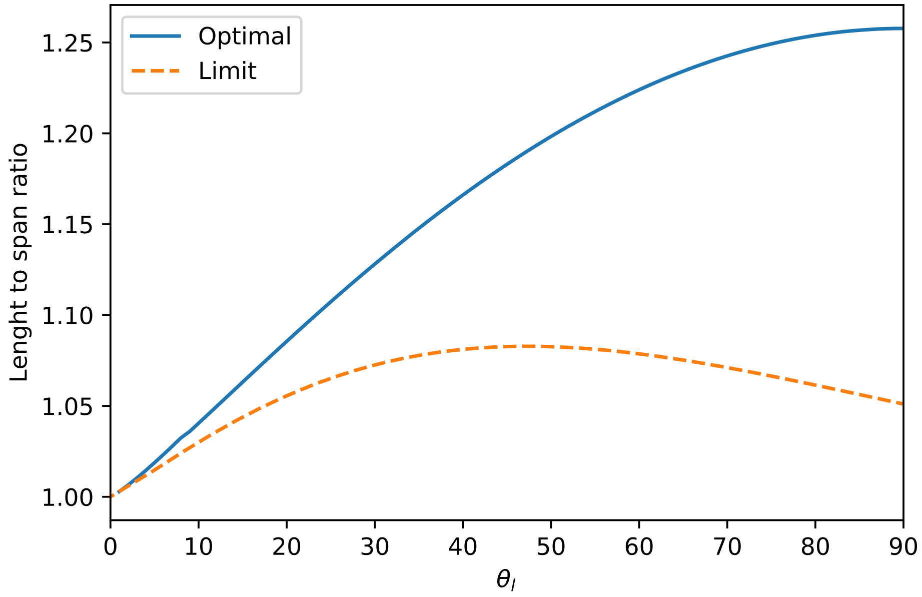


Figure 7.11. Optimal catenary length to span

7.4 DRAG CATENARY MECHANICS IN CARTESIAN SPACE: FINITE DIFFERENCES

7.4.1 STATICS

The model given by (7.17) and (7.18), namely

$$\begin{aligned}\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}s}+\mathbf{q}&=\mathbf{0}\quad\text{and}\\ ||\mathbf{F}||\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}s}&=\mathbf{F},\end{aligned}$$

may be discretised in N segments and $N+1$ nodes. The discretisation of s is given by the sequence $s_j=\{0,s_1,\dots,s_j,\dots,s_N\}$, where $j=0,\dots,N$. Accordingly, variables $\mathbf{F}$ and $\mathbf{r}$ are discretised so that $\mathbf{F}_j=\mathbf{F}(s_j)$ and $\mathbf{r}_j=\mathbf{r}(s_j)$. This finite differences approach discretises the differential equation with respect to the mid-segment using first order centred differences. The derivatives become

$$\begin{aligned}\left.\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}s}\right|_{j-\frac{1}{2}}&\approx\frac{\mathbf{F}_j-\mathbf{F}_{j-1}}{\Delta s_j}\quad\text{and}\\ \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}s}\right|_{j-\frac{1}{2}}&\approx\frac{\mathbf{r}_j-\mathbf{r}_{j-1}}{\Delta s_j},\end{aligned}$$

Also, for the other terms averages are adopted; these become

$$\mathbf{q}_{j-\frac{1}{2}} = \frac{1}{2}\left(\mathbf{q}_{j-1} + \mathbf{q}_j\right), \quad \mathbf{F}_{j-\frac{1}{2}} = \frac{1}{2}\left(\mathbf{F}_{j-1} + \mathbf{F}_j\right), \quad \text{and}$$
$$||\mathbf{F}||_{j-\frac{1}{2}} = \frac{1}{2}\left(T_{j-1} + T_j\right).$$

Please note that the definition $T = ||\mathbf{F}||$ was used here. This discretisation approach is similar to that in the WHOI cable model [53, 54].

Taking this into account, the equations for segment j, with $j = 1, \dots, N$, are given by

$$\begin{aligned}\frac{\mathbf{F}_j - \mathbf{F}_{j-1}}{\Delta s_j} + \frac{1}{2}\left(\mathbf{q}_{j-1} + \mathbf{q}_j\right) =& \mathbf{0} \quad \text{and} \\ \left(T_{j-1} + T_j\right)\frac{\mathbf{r}_j - \mathbf{r}_{j-1}}{\Delta s_j} =& \left(\mathbf{F}_{j-1} + \mathbf{F}_j\right),\end{aligned}$$

where $\Delta s_j = s_j - s_{j-1}$. This gives the set of non-linear equations, for $j = 1, \dots, N$

$$\mathbf{f}_{1j}(\mathbf{F}_{j-1}, \mathbf{r}_{j-1}, \mathbf{F}_j, \mathbf{r}_j) = \frac{\mathbf{F}_j - \mathbf{F}_{j-1}}{\Delta s_j} + \frac{1}{2}\left(\mathbf{q}_{j-1} + \mathbf{q}_j\right) = \mathbf{0} \quad \text{and} \tag{7.30}$$

$$\mathbf{f}_{2j}(\mathbf{F}_{j-1}, \mathbf{r}_{j-1}, \mathbf{F}_j, \mathbf{r}_j) = \left(T_{j-1} + T_j\right)\frac{\mathbf{r}_j - \mathbf{r}_{j-1}}{\Delta s_j} - \left(\mathbf{F}_{j-1} + \mathbf{F}_j\right) = \mathbf{0}. \tag{7.31}$$

Functions $\mathbf{f}_{1j}$ and $\mathbf{f}_{2j}$ in (7.30) and (7.31) are the model's residuals.

7.4.2 DYNAMICS

The dynamics model may be obtained from a linear momentum balance. The model can be written with respect to an inertial frame as

$$\begin{aligned}\frac{\partial \mathbf{F}}{\partial s} + \mathbf{q} =& m\frac{\partial \mathbf{v}}{\partial t} + \mathbf{m}_a \quad \text{and} \\ ||\mathbf{F}||\frac{\partial \mathbf{r}}{\partial s} =& \mathbf{F},\end{aligned}$$

where $\mathbf{F}$ is internal force, $\mathbf{q}$ force field per unit length, $\mathbf{v} = \partial\mathbf{r}/\partial t$ velocity, m mass per unit length, and $\mathbf{m}_a = m_a\mathbf{a}_n$ load due to added mass. More explicitly, the model is

$$\frac{\partial \mathbf{F}}{\partial s} + \mathbf{w} + \mathbf{d} = m\frac{\partial \mathbf{v}}{\partial t} + m_a\mathbf{a}_n \quad \text{and} \tag{7.32}$$

$$||\mathbf{F}||\frac{\partial \mathbf{r}}{\partial s} = \mathbf{F}. \tag{7.33}$$

Using the same discretisation scheme as in the static problem, spatial derivatives are taken with respect to the mid-segment and other quantities are averaged. Consequently,

$$\begin{aligned}\left.\frac{\partial \mathbf{F}}{\partial s}\right|_{j-\frac{1}{2}} \approx& \frac{\mathbf{F}_j - \mathbf{F}_{j-1}}{\Delta s_j} \quad \text{and} \\ \left.\frac{\partial \mathbf{r}}{\partial s}\right|_{j-\frac{1}{2}} \approx& \frac{\mathbf{r}_j - \mathbf{r}_{j-1}}{\Delta s_j},\end{aligned}$$

and non-derivative terms become

$$\mathbf{w}_{j-\frac{1}{2}} = \frac{1}{2}\left(\mathbf{w}_{j-1} + \mathbf{w}_j\right), \quad \mathbf{d}_{j-\frac{1}{2}} = \frac{1}{2}\left(\mathbf{d}_{j-1} + \mathbf{d}_j\right), \quad \left.\frac{\partial \mathbf{v}}{\partial t}\right|_{j-\frac{1}{2}} = \frac{1}{2}\left(\mathbf{a}_{j-1} + \mathbf{a}_j\right),$$

$$\mathbf{a}_{n_{j-\frac{1}{2}}} = \frac{1}{2}\left(\mathbf{a}_{n_{j-1}} + \mathbf{a}_{n_j}\right), \quad \mathbf{F}_{j-\frac{1}{2}} = \frac{1}{2}\left(\mathbf{F}_{j-1} + \mathbf{F}_j\right), \quad \text{and} \quad ||\mathbf{F}||_{j-\frac{1}{2}} = \frac{1}{2}\left(T_{j-1} + T_j\right).$$

Taking this into account, the equations for segment j, with $j = 1, \ldots, N$, are given by

$$\frac{\mathbf{F}_j - \mathbf{F}_{j-1}}{\Delta s_j} + \frac{1}{2}\left(\mathbf{w}_{j-1} + \mathbf{w}_j\right) + \frac{1}{2}\left(\mathbf{d}_{j-1} + \mathbf{d}_j\right) = \frac{1}{2}m_j\left(\mathbf{a}_{j-1} + \mathbf{a}_j\right) + \frac{1}{2}m_{a_j}\left(\mathbf{a}_{n_{j-1}} + \mathbf{a}_{n_j}\right) \quad \text{and}$$

$$\left(T_{j-1} + T_j\right)\frac{\mathbf{r}_j - \mathbf{r}_{j-1}}{\Delta s_j} = \left(\mathbf{F}_{j-1} + \mathbf{F}_j\right).$$

This gives the set of non-linear equations, for $j = 1, \ldots, N$

$$\begin{aligned} \mathbf{f}_{1j}(\mathbf{F}_{j-1}, \mathbf{r}_{j-1}, \mathbf{v}_{j-1}, \mathbf{a}_{j-1}, \mathbf{F}_j, \mathbf{r}_j, \mathbf{v}_j, \mathbf{a}_j) = {}& \\ \frac{1}{2}m_j\left(\mathbf{a}_{j-1} + \mathbf{a}_j\right) &+ \frac{1}{2}m_{a_j}\left(\mathbf{a}_{n_{j-1}} + \mathbf{a}_{n_j}\right) \\ &- \frac{\mathbf{F}_j - \mathbf{F}_{j-1}}{\Delta s_j} - \frac{1}{2}\left(\mathbf{w}_{j-1} - \mathbf{w}_j\right) - \frac{1}{2}\left(\mathbf{d}_{j-1} + \mathbf{d}_j\right) = \mathbf{0} \end{aligned} \tag{7.34}$$

and

$$\mathbf{f}_{2j}(\mathbf{F}_{j-1}, \mathbf{r}_{j-1}, \mathbf{F}_j, \mathbf{r}_j) = \left(T_{j-1} + T_j\right)\frac{\mathbf{r}_j - \mathbf{r}_{j-1}}{\Delta s_j} - \left(\mathbf{F}_{j-1} + \mathbf{F}_j\right) = \mathbf{0}. \tag{7.35}$$

Functions $\mathbf{f}_{1j}$ and $\mathbf{f}_{2j}$ in (7.34) and (7.35) are the model's residuals.

7.4.3 JACOBIANS OF THE DRAG FUNCTION

From all components of static and dynamic models, the computation of drag Jacobians present the most difficulty. A function for drag per unit length, according to (7.13), is

$$\mathbf{d}(\mathbf{r}, \mathbf{F}, \mathbf{v}) = -\frac{1}{2}\rho_{\mathrm{w}} d\big(C_t|\mathbf{v}_t|\mathbf{v}_t + C_n|\mathbf{v}_n|\mathbf{v}_n\big);$$

this model finally depends on $\mathbf{r}$, $\mathbf{F}$, and $\mathbf{v}$. This comes from the way components are defined. Dependence on $\mathbf{F}$ comes from the definition of tangential direction, *i.e.* $\mathbf{e}_t = \mathbf{F}/||\mathbf{F}||$, that is used when defining tangential and normal components of relative velocity, *i.e.*, $\mathbf{v}_t = (\mathbf{v}_r^\intercal \mathbf{e}_t)\mathbf{e}_t$ and $\mathbf{v}_n = (\mathbf{v}_r - \mathbf{v}_r^\intercal \mathbf{e}_t)\mathbf{e}_t$. Dependence on $\mathbf{v}$ comes from the definition of relative velocity, *i.e.*, $\mathbf{v}_r = \mathbf{v} - \mathbf{V}_c^n$. Finally, dependence on $\mathbf{r}$ comes indirectly from the spatial distribution of current velocity, this means that $\mathbf{V}_c^n(\mathbf{r})$. Assuming ρ_{w}, d, C_t, and C_n constant, the Jacobian of $\mathbf{d}$ with respect to $\mathbf{F}$ is

$$\begin{aligned} \mathbf{K}_{\mathbf{dF}} = \frac{\partial \mathbf{d}}{\partial \mathbf{F}} &= \frac{\partial \mathbf{d}}{\partial \mathbf{v}_t}\frac{\partial \mathbf{v}_t}{\partial \mathbf{F}} + \frac{\partial \mathbf{d}}{\partial \mathbf{v}_n}\frac{\partial \mathbf{v}_n}{\partial \mathbf{F}} \\ &= -\frac{1}{2}\rho_{\mathrm{w}} d\left(C_t(v_t\mathbf{I} + \mathbf{v}_t\mathbf{e}_t^\intercal) - C_n(v_n\mathbf{I} + \mathbf{v}_t n\mathbf{e}_n^\intercal)\right)\frac{1}{T}\left(v_t\mathbf{I} + \mathbf{e}_t\mathbf{v}_r^\intercal\right)\left(\mathbf{I} - \mathbf{e}_t\mathbf{e}_t^\intercal\right), \end{aligned}$$

with respect to **r** is

$$\begin{aligned}\mathbf{K}_{\mathbf{dr}} = \frac{\partial \mathbf{d}}{\partial \mathbf{r}} &= \left(\frac{\partial \mathbf{d}}{\partial \mathbf{v}_t}\frac{\partial \mathbf{v}_t}{\partial \mathbf{v}_r} + \frac{\partial \mathbf{d}}{\partial \mathbf{v}_n}\frac{\partial \mathbf{v}_n}{\partial \mathbf{v}_r}\right)\frac{\partial \mathbf{v}_r}{\partial \mathbf{V}_c^n}\frac{\partial \mathbf{V}_c^n}{\partial \mathbf{r}} \\ &= \frac{1}{2}\rho_{\mathrm{w}} d\left(C_t(v_t\mathbf{I} + \mathbf{v}_t\mathbf{e}_t^{\mathsf{T}})\mathbf{e}_t\mathbf{e}_t^{\mathsf{T}} + C_n(v_n\mathbf{I} + \mathbf{v}_t n\mathbf{e}_n^{\mathsf{T}})(\mathbf{I} - \mathbf{e}_t\mathbf{e}_t^{\mathsf{T}})\right)\frac{\partial \mathbf{V}_c^n}{\partial \mathbf{r}},\end{aligned}$$

and with respect to **v** is

$$\begin{aligned}\mathbf{K}_{\mathbf{dv}} = \frac{\partial \mathbf{d}}{\partial \mathbf{v}} &= \left(\frac{\partial \mathbf{d}}{\partial \mathbf{v}_t}\frac{\partial \mathbf{v}_t}{\partial \mathbf{v}_r} + \frac{\partial \mathbf{d}}{\partial \mathbf{v}_n}\frac{\partial \mathbf{v}_n}{\partial \mathbf{v}_r}\right)\frac{\partial \mathbf{v}_r}{\partial \mathbf{v}} \\ &= -\frac{1}{2}\rho_{\mathrm{w}} d\left(C_t(v_t\mathbf{I} + \mathbf{v}_t\mathbf{e}_t^{\mathsf{T}})\mathbf{e}_t\mathbf{e}_t^{\mathsf{T}} + C_n(v_n\mathbf{I} + \mathbf{v}_t n\mathbf{e}_n^{\mathsf{T}})(\mathbf{I} - \mathbf{e}_t\mathbf{e}_t^{\mathsf{T}})\right).\end{aligned}$$

7.4.4 TANGENT LINEAR SYSTEM IN STEADY STATE

Because the static model is non-linear, the assembled residual function may be written as

$$\mathbf{f}(\mathbf{X}) = \mathbf{0},$$

with linear Taylor-series expansion around $\mathbf{X} = \mathbf{X}_0$

$$\mathbf{f}(\mathbf{X}) \approx \mathbf{f}(\mathbf{X}_0) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{X}}\right|_{\mathbf{X}=\mathbf{X}_0}(\mathbf{X} - \mathbf{X}_0) = \mathbf{0}.$$

This lineal approximation can be rewritten as the linear equation system

$$\mathbf{K}_{\mathbf{f}}(\mathbf{X}_0)\mathbf{d}_{\mathbf{X}} = -\mathbf{f}(\mathbf{X}_0),$$

where

$$\mathbf{K}_{\mathbf{f}}(\mathbf{X}_0) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{X}}\right|_{\mathbf{X}=\mathbf{X}_0} \quad \text{and} \quad \mathbf{d}_{\mathbf{X}} = \mathbf{X} - \mathbf{X}_0.$$

The tangent linear system from (7.30) and (7.31) is

$$\begin{bmatrix} -\frac{1}{\Delta s_j}\mathbf{I} + \frac{1}{2}\mathbf{K}_{\mathbf{dF}_{j-1}} & \frac{1}{2}\mathbf{K}_{\mathbf{dr}_{j-1}} & \frac{1}{\Delta s_j}\mathbf{I} + \frac{1}{2}\mathbf{K}_{\mathbf{dF}_j} & \frac{1}{2}\mathbf{K}_{\mathbf{dr}_j} \\ -\mathbf{I} + (\mathbf{r}_j - \mathbf{r}_{j-1})\mathbf{e}_{t_{j-1}}^{\mathsf{T}} & -\frac{T_{j-1}+T_j}{\Delta s_j}\mathbf{I} & -\mathbf{I} + (\mathbf{r}_j - \mathbf{r}_{j-1})\mathbf{e}_{t_j}^{\mathsf{T}} & \frac{T_{j-1}+T_j}{\Delta s_j}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{d}_{\mathbf{F}_{j-1}} \\ \mathbf{d}_{\mathbf{r}_{j-1}} \\ \mathbf{d}_{\mathbf{F}_j} \\ \mathbf{d}_{\mathbf{r}_j} \end{bmatrix} = \begin{bmatrix} -\mathbf{f}_{1j} \\ -\mathbf{f}_{2j} \end{bmatrix}.$$

7.4.5 DYNAMICS TIME INTEGRATION

Newmark-β is a classical algorithm for implicit time integration. This is, from known kinematic conditions $\mathbf{r}_n$, $\mathbf{v}_n$, and $\mathbf{a}_n$, compute kinematic conditions in the next time step $t_{n+1} =$

$t_n + \Delta t$, *i.e.*, $\mathbf{r}_{n+1}$, $\mathbf{v}_{n+1}$, and $\mathbf{a}_{n+1}$. The algorithm is [28]

$$\mathbf{a}_{n+1} = \frac{1}{\beta \Delta t^2}\left(\mathbf{r}_{n+1} - \mathbf{r}_n - \Delta t \mathbf{v}_n\right) - \left(\frac{1}{2\beta} - 1\right)\mathbf{a}_n, \tag{7.36}$$

$$\mathbf{v}_{n+1} = \frac{\gamma}{\beta \Delta t}\left(\mathbf{r}_{n+1} - \mathbf{r}_n\right) - \left(\frac{\gamma}{\beta} - 1\right)\mathbf{v}_n - \Delta t\left(\frac{\gamma}{2\beta} - 1\right)\mathbf{a}_n. \tag{7.37}$$

The tangent linear system from (7.34) and (7.35) is

$$\begin{bmatrix} \mathbf{K}_{1\mathbf{F}_{j-1}} & \mathbf{K}_{1\mathbf{r}_{j-1}} & \mathbf{K}_{1\mathbf{v}_{j-1}} & \mathbf{K}_{1\mathbf{M}_{j-1}} & \mathbf{K}_{1\mathbf{F}_j} & \mathbf{K}_{1\mathbf{r}_j} & \mathbf{K}_{1\mathbf{v}_j} & \mathbf{K}_{1\mathbf{M}_j} \\ \mathbf{K}_{2\mathbf{F}_{j-1}} & \mathbf{K}_{2\mathbf{r}_{j-1}} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{2\mathbf{F}_j} & \mathbf{K}_{2\mathbf{r}_j} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{d}_{\mathbf{F}_{j-1}} \\ \mathbf{d}_{\mathbf{r}_{j-1}} \\ \mathbf{d}_{\mathbf{v}_{j-1}} \\ \mathbf{d}_{\mathbf{a}_{j-1}} \\ \mathbf{d}_{\mathbf{F}_j} \\ \mathbf{d}_{\mathbf{r}_j} \\ \mathbf{d}_{\mathbf{v}_j} \\ \mathbf{d}_{\mathbf{a}_j} \end{bmatrix} = \begin{bmatrix} -\mathbf{f}_{1j} \\ -\mathbf{f}_{2j} \end{bmatrix}$$

where

$$\mathbf{K}_{1\mathbf{F}_{j-1}} = \frac{1}{\Delta s_j}\mathbf{I} - \frac{1}{2}\mathbf{K}_{\mathbf{dF}_{j-1}} + \frac{1}{2}\mathbf{K}_{\mathbf{aF}_{j-1}}, \qquad \mathbf{K}_{1\mathbf{r}_{j-1}} = -\frac{1}{2}\mathbf{K}_{\mathbf{dr}_{j-1}},$$

$$\mathbf{K}_{1\mathbf{v}_{j-1}} = -\frac{1}{2}\mathbf{K}_{\mathbf{dv}_{j-1}}, \qquad \mathbf{K}_{1\mathbf{M}_{j-1}} = \frac{1}{2}m_j\mathbf{I} + \frac{1}{2}m_{a_j}\left(\mathbf{I} - \mathbf{e}_{t_{j-1}}\mathbf{e}_{t_{j-1}}^{\mathsf{T}}\right),$$

$$\mathbf{K}_{1\mathbf{F}_j} = -\frac{1}{\Delta s_j}\mathbf{I} + \frac{1}{2}\mathbf{K}_{\mathbf{F}_{j-1}}, \qquad \mathbf{K}_{1\mathbf{r}_j} = -\frac{1}{2}\mathbf{K}_{\mathbf{dr}_j},$$

$$\mathbf{K}_{1\mathbf{v}_j} = -\frac{1}{2}\mathbf{K}_{\mathbf{dv}_j}, \qquad \mathbf{K}_{1\mathbf{M}_j} = \frac{1}{2}m_j\mathbf{I} + \frac{1}{2}m_{a_j}\left(\mathbf{I} - \mathbf{e}_{t_j}\mathbf{e}_{t_j}^{\mathsf{T}}\right),$$

$$\mathbf{K}_{2\mathbf{F}_{j-1}} = -\mathbf{I} + (\mathbf{r}_j - \mathbf{r}_{j-1})\mathbf{e}_{t_{j-1}}^{\mathsf{T}}, \qquad \mathbf{K}_{2\mathbf{r}_{j-1}} = -\frac{T_{j-1} + T_j}{\Delta s_j}\mathbf{I},$$

$$\mathbf{K}_{2\mathbf{F}_j} = -\mathbf{I} + (\mathbf{r}_j - \mathbf{r}_{j-1})\mathbf{e}_{t_j}^{\mathsf{T}}, \qquad \text{and} \qquad \mathbf{K}_{2\mathbf{r}_j} = \frac{T_{j-1} + T_j}{\Delta s_j}\mathbf{I}.$$

From (7.36) and (7.37)

$$\mathbf{f}_{3j}(\mathbf{r}_{n+1}, \mathbf{a}_{n+1}) = \mathbf{a}_{n+1} - \frac{1}{\beta \Delta t^2}\mathbf{r}_{n+1} + \frac{1}{\beta \Delta t^2}\left(\mathbf{r}_n + \Delta t \mathbf{v}_n\right) + \left(\frac{1}{2\beta} - 1\right)\mathbf{a}_n = \mathbf{0} \quad \text{and}$$

$$\mathbf{f}_{4j}(\mathbf{v}_{n+1}, \mathbf{a}_{n+1}) = \mathbf{v}_{n+1} - \frac{\gamma}{\beta \Delta t}\mathbf{r}_{n+1} + \frac{\gamma}{\beta \Delta t}\mathbf{r}_n + \left(\frac{\gamma}{\beta} - 1\right)\mathbf{v}_n + \Delta t\left(\frac{\gamma}{2\beta} - 1\right)\mathbf{a}_n = \mathbf{0},$$

the corresponding tangent system for the j-th node is

$$\begin{bmatrix} \mathbf{0} & -\frac{1}{\beta \Delta t^2}\mathbf{I} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & -\frac{\gamma}{\beta \Delta t}\mathbf{I} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{d}_{\mathbf{F}_j} \\ \mathbf{d}_{\mathbf{r}_j} \\ \mathbf{d}_{\mathbf{v}_j} \\ \mathbf{d}_{\mathbf{a}_j} \end{bmatrix} = \begin{bmatrix} -\mathbf{f}_{3j} \\ -\mathbf{f}_{4j} \end{bmatrix}.$$

7.5 DRAG CATENARY MECHANICS IN CARTESIAN SPACE: FINITE ELEMENTS

7.5.1 STATICS

The model given by (7.17) and (7.18), namely

$$\begin{aligned}\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}s}+\mathbf{q}=&\mathbf{0}\quad\text{and}\\ ||\mathbf{F}||\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}s}=&\mathbf{F},\end{aligned}$$

can be solved by using a finite element method (FEM) mixed formulation [28]. This means that (7.17) and (7.18) are solved simultaneously for **F** and **r**, and both variables have the same level of discretisation. This FEM formulation holds that similarity to the FD method. Now, writing again the model, but using FEM notation one has

$$\mathbf{F},_s+\mathbf{w}+\mathbf{d}=\mathbf{0}\quad\text{and}\tag{7.38}$$

$$||\mathbf{F}||\mathbf{r},_s-\mathbf{F}=\mathbf{0}.\tag{7.39}$$

Note that in FEM notation $\mathbf{F},_s=\mathrm{d}\mathbf{F}/\mathrm{d}s$ and $\mathbf{r},_s=\mathrm{d}\mathbf{r}/\mathrm{d}s$. Also note that load **q** is written explicitly as a sum of weight and drag, *i.e.*, $\mathbf{w}+\mathbf{d}$. This distinction is important because **w** is assumed as a constant field over each cable segment and **d** variable. Equations (7.38) and (7.39) are named the *strong form* of the model.

Using Galerkin formulation of weighted residuals, (arbitrary) weight functions $\mathbf{w}_1(s)$ and $\mathbf{w}_2(s)$ are first defined to dot-multiply both sides of the equation and then integrate over the domain $s\in[0,L]$. This gives

$$\begin{aligned}\int_0^L(\mathbf{F},_s+\mathbf{w}+\mathbf{d})\cdot\mathbf{w}_1(s)\mathrm{d}s=&0\quad\text{and}\\ \int_0^L(||\mathbf{F}||\mathbf{r},_s-\mathbf{F})\cdot\mathbf{w}_2(s)\mathrm{d}s=&0.\end{aligned}$$

These integrals may be solved term-wise, *i.e.*,

$$\begin{aligned}\int_0^L\mathbf{F},_s\cdot\mathbf{w}_1(s)\mathrm{d}s+\int_0^L\mathbf{w}\cdot\mathbf{w}_1(s)\mathrm{d}s+\int_0^L\mathbf{d}\cdot\mathbf{w}_1(s)\mathrm{d}s=&0\quad\text{and}\\ \int_0^L||\mathbf{F}||\mathbf{r},_s\cdot\mathbf{w}_2(s)\mathrm{d}s-\int_0^L\mathbf{F}\cdot\mathbf{w}_2(s)\mathrm{d}s=&0.\end{aligned}$$

Finally, using integration by parts on the first equation, one obtains the static model in its *weak form*, namely

$$\mathbf{F}(s)\cdot\mathbf{w}_1(s)|_0^L-\int_0^L\mathbf{F}\cdot\mathbf{w}_1,_s\mathrm{d}s+\int_0^L\mathbf{w}\cdot\mathbf{w}_1(s)\mathrm{d}s+\int_0^L\mathbf{d}\cdot\mathbf{w}_1(s)\mathrm{d}s=0\quad\text{and}\tag{7.40}$$

$$\int_0^L||\mathbf{F}||\mathbf{r},_s\cdot\mathbf{w}_2(s)\mathrm{d}s-\int_0^L\mathbf{F}\cdot\mathbf{w}_2(s)\mathrm{d}s=0.\tag{7.41}$$

This problem may be discretised in N segments and $N+1$ nodes. The discretisation of s is given by the sequence $\{s_1, \dots, s_j, \dots, s_{N+1}\}$. This allows one to write a discretised version of (7.40) and (7.41), namely

$$\mathbf{F}(s) \cdot \mathbf{w}_1(s)|_0^L + \sum_{j=1}^{N} \left(-\int_{s_j}^{s_{j+1}} \mathbf{F} \cdot \mathbf{w}_{1,s} \mathrm{d}s + \int_{s_j}^{s_{j+1}} \mathbf{w} \cdot \mathbf{w}_1(s) \mathrm{d}s + \int_{s_j}^{s_{j+1}} \mathbf{d} \cdot \mathbf{w}_1(s) \mathrm{d}s \right) = 0 \tag{7.42}$$

$$\text{and} \quad \sum_{j=1}^{N} \left(\int_{s_j}^{s_{j+1}} ||\mathbf{F}|| \mathbf{r}_{,s} \cdot \mathbf{w}_2(s) \mathrm{d}s - \int_{s_j}^{s_{j+1}} \mathbf{F} \cdot \mathbf{w}_2(s) \mathrm{d}s \right) = 0. \tag{7.43}$$

Variables $\mathbf{F}$, $\mathbf{r}$, $\mathbf{w}_1$, and $\mathbf{w}_2$ are discretised using linear Lagrange polynomials. This means, using $\mathbf{F}(s)$ as an example, that in $s \in [s_j, s_{j+1}]$

$$\mathbf{F}(s) = \mathrm{N}_1^j(s)\mathbf{F}_j + \mathrm{N}_2^j(s)\mathbf{F}_{j+1} = \begin{bmatrix} \mathrm{N}_1^j(s) & \mathrm{N}_2^j(s) \end{bmatrix} \begin{bmatrix} \mathbf{F}_j \\ \mathbf{F}_{j+1} \end{bmatrix},$$

where

$$\begin{aligned} \mathrm{N}_1^j(s) =& \frac{s_{j+1} - s}{s_{j+1} - s_j} = \frac{s_{j+1} - s}{\Delta s_j} \quad \text{and} \\ \mathrm{N}_2^j(s) =& \frac{s - s_j}{s_{j+1} - s_j} = \frac{s - s_j}{\Delta s_j}. \end{aligned}$$

Moreover, the derivatives become

$$\mathbf{F}_{,s}(s) = \mathrm{N}_1'^j(s)\mathbf{F}_j + \mathrm{N}_2'^j(s)\mathbf{F}_{j+1} = \begin{bmatrix} \mathrm{N}_1'^j(s) & \mathrm{N}_2'^j(s) \end{bmatrix} \begin{bmatrix} \mathbf{F}_j \\ \mathbf{F}_{j+1} \end{bmatrix},$$

where

$$\mathrm{N}_1'^j(s) = -\frac{1}{\Delta s_j} \quad \text{and} \quad \mathrm{N}_2'^j(s) = \frac{1}{\Delta s_j}.$$

Taking this into account, the integral terms in (7.42) and (7.43) are as follows. For the internal forces term one has

$$\begin{aligned} \int_{s_j}^{s_{j+1}} \mathbf{F} \cdot \mathbf{w}_{1,s} \mathrm{d}s =& \int_{s_j}^{s_{j+1}} \left(\mathrm{N}_1^j(s)\mathbf{F}_j + \mathrm{N}_2^j(s)\mathbf{F}_{j+1} \right) \cdot \left(\mathrm{N}_1'^j(s)\mathbf{w}_{1_j} + \mathrm{N}_2'^j(s)\mathbf{w}_{1_{j+1}} \right) \mathrm{d}s \\ =& \begin{bmatrix} \mathbf{w}_{1_j}^\mathsf{T} & \mathbf{w}_{1_{j+1}}^\mathsf{T} \end{bmatrix} \int_{s_j}^{s_{j+1}} \begin{bmatrix} \mathrm{N}_1'^j(s) \\ \mathrm{N}_2'^j(s) \end{bmatrix} \begin{bmatrix} \mathrm{N}_1^j(s) & \mathrm{N}_2^j(s) \end{bmatrix} \mathrm{d}s \begin{bmatrix} \mathbf{F}_j \\ \mathbf{F}_{j+1} \end{bmatrix} \\ =& \begin{bmatrix} \mathbf{w}_{1_j}^\mathsf{T} & \mathbf{w}_{1_{j+1}}^\mathsf{T} \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \mathbf{F}_j \\ \mathbf{F}_{j+1} \end{bmatrix}. \end{aligned}$$

For the weight term one assumes that $\mathbf{w}$ is constant for each segment, *i.e.*, $\mathbf{w} = \mathbf{w}_j$, for $s \in [s_j, s_{j+1}]$. Then, the expression becomes

$$\int_{s_j}^{s_{j+1}} \mathbf{w} \cdot \mathbf{w}_1(s)\mathrm{d}s = \int_{s_j}^{s_{j+1}} \mathbf{w}_j \cdot \left(\mathrm{N}_1^j(s)\mathbf{w}_{1_j} + \mathrm{N}_2^j(s)\mathbf{w}_{1_{j+1}}\right) \mathrm{d}s = \begin{bmatrix} \mathbf{w}_{1_j}^\intercal & \mathbf{w}_{1_{j+1}}^\intercal \end{bmatrix} \begin{bmatrix} \frac{\Delta s_j}{2}\mathbf{w}_j \\ \frac{\Delta s_j}{2}\mathbf{w}_j \end{bmatrix}.$$

For the drag term one has

$$\begin{aligned} \int_{s_j}^{s_{j+1}} \mathbf{d} \cdot \mathbf{w}_1(s)\mathrm{d}s &= \int_{s_j}^{s_{j+1}} \left(\mathrm{N}_1^j(s)\mathbf{d}_j + \mathrm{N}_2^j(s)\mathbf{d}_{j+1}\right) \cdot \left(\mathrm{N}_1^j(s)\mathbf{w}_{1_j} + \mathrm{N}_2^j(s)\mathbf{w}_{1_{j+1}}\right) \mathrm{d}s \\ &= \begin{bmatrix} \mathbf{w}_{1_j}^\intercal & \mathbf{w}_{1_{j+1}}^\intercal \end{bmatrix} \begin{bmatrix} \frac{\Delta s_j}{3}\mathbf{d}_j + \frac{\Delta s_j}{6}\mathbf{d}_{j+1} \\ \frac{\Delta s_j}{6}\mathbf{d}_j + \frac{\Delta s_j}{3}\mathbf{d}_{j+1} \end{bmatrix}. \end{aligned}$$

For the force-magnitude-direction term, force magnitude is defined as $||\mathbf{F}|| = T(s)$, and for a segment $T(s) = \mathrm{N}_1^j(s)T_j + \mathrm{N}_2^j(s)T_{j+1}$; thus

$$\begin{aligned} &\int_{s_j}^{s_{j+1}} ||\mathbf{F}||\mathbf{r},_s \cdot \mathbf{w}_2(s)\mathrm{d}s \\ &= \int_{s_j}^{s_{j+1}} \left(\mathrm{N}_1^j(s)T_j + \mathrm{N}_2^j(s)T_{j+1}\right) \left(\mathrm{N}_1'^j(s)\mathbf{r}_j + \mathrm{N}_2'^j(s)\mathbf{r}_{j+1}\right) \cdot \left(\mathrm{N}_1^j(s)\mathbf{w}_{2_j} + \mathrm{N}_2^j(s)\mathbf{w}_{2_{j+1}}\right) \mathrm{d}s \\ &= \begin{bmatrix} \mathbf{w}_{2_j}^\intercal & \mathbf{w}_{2_{j+1}}^\intercal \end{bmatrix} \begin{bmatrix} -\frac{1}{3}T_j - \frac{1}{6}T_{j+1} & \frac{1}{3}T_j + \frac{1}{6}T_{j+1} \\ -\frac{1}{6}T_j - \frac{1}{3}T_{j+1} & \frac{1}{6}T_j + \frac{1}{3}T_{j+1} \end{bmatrix} \begin{bmatrix} \mathbf{r}_j \\ \mathbf{r}_{j+1} \end{bmatrix}. \end{aligned}$$

The force term becomes

$$\begin{aligned} \int_{s_j}^{s_{j+1}} \mathbf{F} \cdot \mathbf{w}_2(s)\mathrm{d}s &= \int_{s_j}^{s_{j+1}} \left(\mathrm{N}_1^j(s)\mathbf{F}_j + \mathrm{N}_2^j(s)\mathbf{F}_{j+1}\right) \cdot \left(\mathrm{N}_1^j(s)\mathbf{w}_{2_j} + \mathrm{N}_2^j(s)\mathbf{w}_{2_{j+1}}\right) \mathrm{d}s \\ &= \begin{bmatrix} \mathbf{w}_{2_j}^\intercal & \mathbf{w}_{2_{j+1}}^\intercal \end{bmatrix} \begin{bmatrix} \frac{\Delta s_j}{3} & \frac{\Delta s_j}{6} \\ \frac{\Delta s_j}{6} & \frac{\Delta s_j}{3} \end{bmatrix} \begin{bmatrix} \mathbf{F}_j \\ \mathbf{F}_{j+1} \end{bmatrix}. \end{aligned}$$

The discretised version of (7.42) and (7.43) is

$$\begin{aligned} \mathbf{F}(s) \cdot \mathbf{w}_1(s)|_0^L + \sum_{j=1}^{N} \begin{bmatrix} \mathbf{w}_{1_j}^\intercal & \mathbf{w}_{1_{j+1}}^\intercal \end{bmatrix} &\left(- \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \mathbf{F}_j \\ \mathbf{F}_{j+1} \end{bmatrix} \right. \\ &\left. + \begin{bmatrix} \frac{\Delta s_j}{2}\mathbf{w}_j \\ \frac{\Delta s_j}{2}\mathbf{w}_j \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_j}{3}\mathbf{d}_j + \frac{\Delta s_j}{6}\mathbf{d}_{j+1} \\ \frac{\Delta s_j}{6}\mathbf{d}_j + \frac{\Delta s_j}{3}\mathbf{d}_{j+1} \end{bmatrix} \right) = 0 \quad (7.44) \end{aligned}$$

and

$$\begin{aligned} \sum_{j=1}^{N} \begin{bmatrix} \mathbf{w}_{2_j}^\intercal & \mathbf{w}_{2_{j+1}}^\intercal \end{bmatrix} &\left(\begin{bmatrix} -\frac{1}{3}T_j - \frac{1}{6}T_{j+1} & \frac{1}{3}T_j + \frac{1}{6}T_{j+1} \\ -\frac{1}{6}T_j - \frac{1}{3}T_{j+1} & \frac{1}{6}T_j + \frac{1}{3}T_{j+1} \end{bmatrix} \begin{bmatrix} \mathbf{r}_j \\ \mathbf{r}_{j+1} \end{bmatrix} \right. \\ &\left. - \begin{bmatrix} \frac{\Delta s_j}{3} & \frac{\Delta s_j}{6} \\ \frac{\Delta s_j}{6} & \frac{\Delta s_j}{3} \end{bmatrix} \begin{bmatrix} \mathbf{F}_j \\ \mathbf{F}_{j+1} \end{bmatrix} \right) = 0. \quad (7.45) \end{aligned}$$

7.5.2 DYNAMICS

The dynamic model in (7.32) and (7.33), namely

$$\begin{aligned}\frac{\partial \mathbf{F}}{\partial s} + \mathbf{w} + \mathbf{d} =& m\frac{\partial \mathbf{v}}{\partial t} + m_a\mathbf{a}_n \quad \text{and}\\ ||\mathbf{F}||\frac{\partial \mathbf{r}}{\partial s} =& \mathbf{F}.\end{aligned}$$

can be solved as well by using a FEM mixed formulation. This means that (7.32) and (7.33) are solved simultaneously for $\mathbf{F}$ and kinematic variables $\mathbf{r}$ and $\mathbf{v}$. In FEM notation one has

$$\mathbf{F},_s + \mathbf{w} + \mathbf{d} - m_a\mathbf{a}_n - m\dot{\mathbf{v}} = \mathbf{0} \quad \text{and} \tag{7.46}$$

$$||\mathbf{F}||\mathbf{r},_s - \mathbf{F} = \mathbf{0}. \tag{7.47}$$

Equations (7.38) and (7.39) are the *strong form* of the dynamic model.

For using Galerkin formulation of weighted residuals, weight functions $\mathbf{w}_1(s)$ and $\mathbf{w}_2(s)$ are defined. Dot-multiplying both sides of the equation by $\mathbf{w}_1(s)$ and $\mathbf{w}_2(s)$ and then integrating over the domain $s \in [0, L]$ gives

$$\begin{aligned}\int_0^L (\mathbf{F},_s + \mathbf{w} + \mathbf{d} - m_a\mathbf{a}_n - m\dot{\mathbf{v}}) \cdot \mathbf{w}_1(s)\mathrm{d}s =& 0 \quad \text{and}\\ \int_0^L (||\mathbf{F}||\mathbf{r},_s - \mathbf{F}) \cdot \mathbf{w}_2(s)\mathrm{d}s =& 0.\end{aligned}$$

Using integration by parts in the first equation and writing the integrals term-wise, the *weak form* becomes

$$\begin{aligned}\mathbf{F}(s)\cdot\mathbf{w}_1(s)|_0^L - \int_0^L \mathbf{F}\cdot\mathbf{w}_1,_s\mathrm{d}s + \int_0^L \mathbf{w}\cdot\mathbf{w}_1(s)\mathrm{d}s + \int_0^L \mathbf{d}\cdot\mathbf{w}_1(s)\mathrm{d}s\\ - \int_0^L m_a\mathbf{a}_n\cdot\mathbf{w}_1(s)\mathrm{d}s - \int_0^L m\dot{\mathbf{v}}\cdot\mathbf{w}_1(s)\mathrm{d}s = 0\end{aligned} \tag{7.48}$$

and

$$\int_0^L ||\mathbf{F}||\mathbf{r},_s\cdot\mathbf{w}_2(s)\mathrm{d}s - \int_0^L \mathbf{F}\cdot\mathbf{w}_2(s)\mathrm{d}s = 0. \tag{7.49}$$

As before, this problem may be discretised in N segments and $N+1$ nodes. The discretised version of (7.40) and (7.41) becomes

$$\begin{aligned}\mathbf{F}(s)\cdot\mathbf{w}_1(s)|_0^L + \sum_{j=1}^{N}\Bigg(- \int_{s_j}^{s_{j+1}} \mathbf{F}\cdot\mathbf{w}_1,_s\mathrm{d}s + \int_{s_j}^{s_{j+1}} \mathbf{w}\cdot\mathbf{w}_1(s)\mathrm{d}s + \int_{s_j}^{s_{j+1}} \mathbf{d}\cdot\mathbf{w}_1(s)\mathrm{d}s\\ - \int_{s_j}^{s_{j+1}} m_a\mathbf{a}_n\cdot\mathbf{w}_1(s)\mathrm{d}s - \int_{s_j}^{s_{j+1}} m\dot{\mathbf{v}}\cdot\mathbf{w}_1(s)\mathrm{d}s\Bigg) = 0\end{aligned} \tag{7.50}$$

and

$$\sum_{j=1}^{N}\left(\int_{s_j}^{s_{j+1}} ||\mathbf{F}||\mathbf{r},_s \cdot \mathbf{w}_2(s)\mathrm{d}s - \int_{s_j}^{s_{j+1}} \mathbf{F}\cdot\mathbf{w}_2(s)\mathrm{d}s\right) = 0. \tag{7.51}$$

Variables $\mathbf{F}$, $\mathbf{r}$, $\mathbf{w}_1$, and $\mathbf{w}_2$ are discretised similarly to the static case, using linear Lagrange polynomials. Consequently, all but inertial terms of (7.50) and (7.50) are solved as in the static case. The additional terms are added mass given by

$$\begin{aligned}\int_{s_j}^{s_{j+1}} m_a\mathbf{a}_n\cdot\mathbf{w}_1(s)\mathrm{d}s &= \int_{s_j}^{s_{j+1}} m_{a_j}\left(\mathrm{N}_1^j(s)\mathbf{a}_{n_j} + \mathrm{N}_2^j(s)\mathbf{a}_{nj+1}\right)\cdot\left(\mathrm{N}_1^j(s)\mathbf{w}_{1_j} + \mathrm{N}_2^j(s)\mathbf{w}_{1_{j+1}}\right)\mathrm{d}s \\ &= \left[\begin{array}{cc}\mathbf{w}_{1_j}^{\mathsf{T}} & \mathbf{w}_{1_{j+1}}^{\mathsf{T}}\end{array}\right] m_{a_j}\left[\begin{array}{c}\frac{\Delta s_j}{3}\mathbf{a}_{n_j} + \frac{\Delta s_j}{6}\mathbf{a}_{n_{j+1}} \\ \frac{\Delta s_j}{6}\mathbf{a}_{n_j} + \frac{\Delta s_j}{3}\mathbf{a}_{n_{j+1}}\end{array}\right]\end{aligned}$$

and inertia by

$$\begin{aligned}\int_{s_j}^{s_{j+1}} m\dot{\mathbf{v}}\cdot\mathbf{w}_1(s)\mathrm{d}s &= \int_{s_j}^{s_{j+1}} m_j\left(\mathrm{N}_1^j(s)\dot{\mathbf{v}}_j + \mathrm{N}_2^j(s)\dot{\mathbf{v}}_{j+1}\right)\cdot\left(\mathrm{N}_1^j(s)\mathbf{w}_{1_j} + \mathrm{N}_2^j(s)\mathbf{w}_{1_{j+1}}\right)\mathrm{d}s \\ &= \left[\begin{array}{cc}\mathbf{w}_{1_j}^{\mathsf{T}} & \mathbf{w}_{1_{j+1}}^{\mathsf{T}}\end{array}\right] m_j\int_{s_j}^{s_{j+1}}\left[\begin{array}{c}\mathrm{N}_1^j(s) \\ \mathrm{N}_2^j(s)\end{array}\right]\left[\begin{array}{cc}\mathrm{N}_1^j(s) & \mathrm{N}_2^j(s)\end{array}\right]\mathrm{d}s\left[\begin{array}{c}\dot{\mathbf{v}}_j \\ \dot{\mathbf{v}}_{j+1}\end{array}\right] \\ &= \left[\begin{array}{cc}\mathbf{w}_{1_j}^{\mathsf{T}} & \mathbf{w}_{1_{j+1}}^{\mathsf{T}}\end{array}\right] m_j\Delta s_j\left[\begin{array}{cc}\frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3}\end{array}\right]\left[\begin{array}{c}\dot{\mathbf{v}}_j \\ \dot{\mathbf{v}}_{j+1}\end{array}\right].\end{aligned}$$

The discretised version of (7.50) and (7.51) is

$$\begin{aligned}&\mathbf{F}(s)\cdot\mathbf{w}_1(s)|_0^L + \sum_{j=1}^{N}\left[\begin{array}{cc}\mathbf{w}_{1_j}^{\mathsf{T}} & \mathbf{w}_{1_{j+1}}^{\mathsf{T}}\end{array}\right]\left(-\left[\begin{array}{cc}-\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2}\end{array}\right]\left[\begin{array}{c}\mathbf{F}_j \\ \mathbf{F}_{j+1}\end{array}\right] + \left[\begin{array}{c}\frac{\Delta s_j}{2}\mathbf{w}_j \\ \frac{\Delta s_j}{2}\mathbf{w}_j\end{array}\right]\right. \\ &\left. + \left[\begin{array}{c}\frac{\Delta s_j}{3}\mathbf{d}_j + \frac{\Delta s_j}{6}\mathbf{d}_{j+1} \\ \frac{\Delta s_j}{6}\mathbf{d}_j + \frac{\Delta s_j}{3}\mathbf{d}_{j+1}\end{array}\right] - m_{a_j}\Delta s_j\left[\begin{array}{c}\frac{1}{3}\mathbf{a}_{n_j} + \frac{1}{6}\mathbf{a}_{n_{j+1}} \\ \frac{1}{6}\mathbf{a}_{n_j} + \frac{1}{3}\mathbf{a}_{n_{j+1}}\end{array}\right] - m_j\Delta s_j\left[\begin{array}{cc}\frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3}\end{array}\right]\left[\begin{array}{c}\dot{\mathbf{v}}_j \\ \dot{\mathbf{v}}_{j+1}\end{array}\right]\right) = 0\end{aligned} \tag{7.52}$$

and

$$\begin{aligned}\sum_{j=1}^{N}\left[\begin{array}{cc}\mathbf{w}_{2_j}^{\mathsf{T}} & \mathbf{w}_{2_{j+1}}^{\mathsf{T}}\end{array}\right]&\left(\left[\begin{array}{cc}-\frac{1}{3}T_j - \frac{1}{6}T_{j+1} & \frac{1}{3}T_j + \frac{1}{6}T_{j+1} \\ -\frac{1}{6}T_j - \frac{1}{3}T_{j+1} & \frac{1}{6}T_j + \frac{1}{3}T_{j+1}\end{array}\right]\left[\begin{array}{c}\mathbf{r}_j \\ \mathbf{r}_{j+1}\end{array}\right]\right. \\ &\left. - \left[\begin{array}{cc}\frac{\Delta s_j}{3} & \frac{\Delta s_j}{6} \\ \frac{\Delta s_j}{6} & \frac{\Delta s_j}{3}\end{array}\right]\left[\begin{array}{c}\mathbf{F}_j \\ \mathbf{F}_{j+1}\end{array}\right]\right) = 0.\end{aligned} \tag{7.53}$$

7.6 DYNAMICS AND LINEAR ELASTICITY USING CABLE FRAME COORDINATES

7.6.1 STRAIN KINEMATICS

All variables are represented as a function of $s \in [0, L]$, the Lagrangian coordinate that represents the non-stretched length of the cable. When the cable is stretched, the length is represented by $p \in [0, L_s]$. The relationship of these two variables is given by the strain ϵ (see *e.g.* [13, 62]), such that

$$\frac{\partial p}{\partial s} = 1 + \epsilon \tag{7.54}$$

Throughout the whole study linear materials are assumed, this means that strain is related to material properties and load by Hooke's law $\epsilon = \frac{T}{EA}$, where T is tension, E is Young's modulus, $A = \pi \mathrm{d}^2/4$ is transversal area, and d is the diameter.

A point on the cable is given at an inertial reference frame as $\mathbf{r} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$. The variation of the cable's position along the stretched coordinate p is given by

$$\frac{\partial \mathbf{r}}{\partial p} = \begin{bmatrix} \frac{\partial x}{\partial p} & \frac{\partial y}{\partial p} & \frac{\partial z}{\partial p} \end{bmatrix}^\mathsf{T}.$$

This derivative gives the tangential direction vector of the cable $\mathbf{e}_t$, such that

$$\mathbf{e}_t = \frac{\partial \mathbf{r}}{\partial p}. \tag{7.55}$$

Then, variation of the cable's position along the non-stretched coordinate s is given by

$$\frac{\partial \mathbf{r}}{\partial s} = \frac{\partial \mathbf{r}}{\partial p}\frac{\mathrm{d}p}{\mathrm{d}s},$$

which combined with (7.54) and (7.55) gives the expression

$$\frac{\partial \mathbf{r}}{\partial s} = \mathbf{e}_t(1+\epsilon). \tag{7.56}$$

7.6.2 STATICS EQUATIONS

The forces equilibrium equation can be written as [62]

$$\frac{\mathrm{D}\mathbf{F}}{\mathrm{D}p} + \mathbf{q} = \mathbf{0}. \tag{7.57}$$

This model indicates that variation on the internal load along the cable length is due to the action of the external loads. Given that $\frac{\mathrm{D}\mathbf{F}}{\mathrm{D}s} = \frac{\mathrm{D}\mathbf{F}}{\mathrm{D}p}\frac{\mathrm{d}p}{\mathrm{d}s}$ and combining (7.54) and (7.57), the forces model becomes

$$\frac{\mathrm{D}\mathbf{F}}{\mathrm{D}s} + \mathbf{q}(1+\epsilon) = \mathbf{0}. \tag{7.58}$$

Moreover, the complete derivative in (7.58), using the definition in (7.7), can be further detailed as

$$\frac{\partial \mathbf{F}}{\partial s} + \boldsymbol{\Omega} \times \mathbf{F} + \mathbf{q}(1+\epsilon) = \mathbf{0}. \tag{7.59}$$

The moments equation is assumed as

$$\frac{\mathrm{D}\mathbf{M}}{\mathrm{D}p} + \frac{\partial \mathbf{r}}{\partial p} \times \mathbf{F} + \mathbf{m}_e = \mathbf{0}, \tag{7.60}$$

where $\mathbf{M}$ is the internal moments vector and $\mathbf{m}_e$ stands for externally applied moments for unit length. If there are not externally applied moments ($\mathbf{m}_e = \mathbf{0}$), it is considered that $\frac{\mathrm{D}\mathbf{M}}{\mathrm{D}s} = \frac{\mathrm{D}\mathbf{M}}{\mathrm{D}p}\frac{\mathrm{d}p}{\mathrm{d}s}$, the complete derivative is expanded as in (7.7), and the definition in (7.55) is considered, then (7.60) becomes

$$\frac{\partial \mathbf{M}}{\partial s} + \boldsymbol{\Omega} \times \mathbf{M} + \mathbf{e}_t \times \mathbf{F}(1+\epsilon) = \mathbf{0}, \tag{7.61}$$

Moment equation becomes important when bending is considered (torsion as well, but is not included here).

7.6.3 DYNAMICS EQUATIONS

The forces equation can be written as [62]

$$\mathbf{m}\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = \frac{\mathrm{D}\mathbf{F}}{\mathrm{D}p} + \mathbf{q}, \tag{7.62}$$

where $\mathbf{m}$ may represent either the mass per unit length or a mass matrix with different mass-per-unit-length components and $\mathbf{v}$ represents velocity. Mass $\mathbf{m}$ should be defined as a matrix when considering added mass.

This model indicates that variation on the internal load along the cable length is due to the action of the external loads. Given that $\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}s} = \frac{\mathrm{D}\mathbf{F}}{\mathrm{D}p}\frac{\mathrm{d}p}{\mathrm{d}s}$ and combining (7.54) and (7.62), the forces model becomes

$$\mathbf{m}\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = \frac{\mathrm{D}\mathbf{F}}{\mathrm{D}s} + \mathbf{q}(1+\epsilon). \tag{7.63}$$

Moreover, the complete derivative in (7.63), using the definition in (7.7), can be further detailed as

$$\mathbf{m}\left(\frac{\partial \mathbf{v}}{\partial t} + \omega \times \mathbf{v}\right) = \frac{\partial \mathbf{F}}{\partial s} + \boldsymbol{\Omega} \times \mathbf{F} + \mathbf{q}(1+\epsilon). \tag{7.64}$$

Throughout this analysis moments are considered static, hence the model is the same as in Eq. (7.61).

7.6.4 COMPATIBILITY EQUATIONS

The compatibility equations arise from assuming a continuous cable, this means that it is valid to write

$$\frac{\partial}{\partial t}\frac{\mathrm{D}\mathbf{r}}{\mathrm{D}s} = \frac{\mathrm{D}}{\mathrm{D}s}\frac{\partial \mathbf{r}}{\partial t}.$$

From this expression and using (7.56) it follows

$$\frac{\partial}{\partial t}\left((1+\epsilon)\hat{t}\right) = \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}s}.$$

If this expression is further developed the following additional system of equation is obtained

$$(1+\epsilon)\omega_2 = \frac{\partial u}{\partial s} + w\Omega_2 - v\Omega_3, \tag{7.65a}$$

$$-(1+\epsilon)\omega_1 = \frac{\partial v}{\partial s} + u\Omega_3 - w\Omega_1, \quad \text{and} \tag{7.65b}$$

$$\frac{\partial \epsilon}{\partial t} = \frac{\partial w}{\partial s} + v\Omega_1 - u\Omega_2. \tag{7.65c}$$

7.7 STATICS COMPUTATIONS USING CABLE COORDINATES

In this section the difference among different two-dimensional models is taken into account. The following modelling scenarios are considered:

- Inextensible cable subject to weight only.
- Extensible cable subject to weight only.
- Inextensible cable subject to weight and drag.
- Extensible cable subject to weight and drag.
- Inextensible cable subject to weight and drag, including bending effects.
- Extensible cable subject to weight and drag, including bending effects.

7.7.1 NON-DIMENSIONAL VARIABLES

Calculation of non-dimensional variables is based on two main scaling quantities: the cable's non-stretched length L and a measure of the total external load per unit length applied to the cable q, adding weight and drag effects. The use of the former quantity is quite usual; regarding the use of the latter quantity, it is more common to scale loads based on weight only (see for instance [62]). Given that it is convenient to study neutrally buoyant cables as well, using a measure of the total load helps one to analyse the relative importance between weight and drag.

The total load per unit length is given in (7.11), then a measure of the total load is

$$q = \bar{w} + \bar{d}, \tag{7.66}$$

where $\bar{w}$ is the cable's weight in water per unit length and $\bar{d}$ is a metric of the total drag per unit length. Such drag may be defined as

$$\bar{d} = \frac{1}{2}\rho_{\mathrm{w}} d C_d \bar{V}^2, \tag{7.67}$$

where $\bar{V}$ is a reference velocity. Two possibilities for this velocity, based on a current profile, are

$$\bar{V} = \max(||V_c(s)||) \quad \text{or} \quad \bar{V} = \sqrt{\left|\left|\frac{1}{L}\int_0^L |V_c(s)|^{\intercal} V_c(s) ds\right|\right|}. \tag{7.68}$$

Application of the scaling quantities L and q gives the following non-dimensional variables:

- Distance-based variables are scaled as $\hat{s} = s/L$, $\hat{x} = x/L$, $\hat{y} = y/L$, and $\hat{z} = z/L$.
- Rotation-related variables are scaled, for $i = 1, 2, \text{and } 3$, as $\hat{\Omega}_i = L\Omega_i$, $\frac{\mathrm{d}\hat{\Omega}_i}{\mathrm{d}\hat{s}} = L^2 \frac{\mathrm{d}\Omega_i}{\mathrm{d}s}$, and $\frac{\mathrm{d}^2\hat{\Omega}_i}{\mathrm{d}\hat{s}^2} = L^3 \frac{\mathrm{d}^2\Omega_i}{\mathrm{d}s^2}$.
- Tension is scaled as $\hat{T} = \frac{T}{qL}$.
- Weight and drag per unit length are respectively scaled as $\hat{w} = \frac{\bar{w}}{q}$ and $\hat{d} = \frac{\bar{d}}{q}$. This gives way to the property $\hat{w} + \hat{d} = 1$.
- Axial and bending flexibility are respectively scaled as $c_s = \frac{qL}{EA}$ and $c_b = \frac{qL^3}{EI}$. Following this, strain may be defined as function of tension as $\epsilon = c_s\hat{T}$.
- Velocities are scaled using the reference velocity such that for any velocity component $\hat{v} = v/\bar{V}$.

7.7.2 DIFFERENT MODELS IN TWO DIMENSIONS

All the studied cases are two-dimensional. This means that the following simplifications are always applied:

- In the transformation (7.3), the rotation around the $x-$axis is eliminated, this means that $\phi = 0$.
- Current profiles and, consequently, drag loads are restricted to the $xz-$plane.

These assumptions have further consequences. The equations for cable element's coordinates (7.4)–(7.6) becomes

$$\frac{\partial x}{\partial s} = \sin\theta(1+\epsilon), \tag{7.69a}$$

$$\frac{\partial y}{\partial s} = 0, \quad \text{and} \tag{7.69b}$$

$$\frac{\partial z}{\partial s} = \cos\theta(1+\epsilon). \tag{7.69c}$$

The rotation vector in (7.8) becomes

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 \\ \Omega \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial\theta}{\partial s} \\ 0 \end{bmatrix}. \tag{7.70}$$

The weight vector in (7.12) becomes

$$\mathbf{w} = \bar{w} \begin{bmatrix} -\sin\theta \\ 0 \\ \cos\theta \end{bmatrix}. \tag{7.71}$$

The drag vector in (7.15) becomes

$$\mathbf{d} = -\frac{1}{2}\rho_{\mathrm{w}} d C_d \begin{bmatrix} f_n(\varphi) v_n |v_n| \\ 0 \\ f_t(\varphi) v_t |v_t| \end{bmatrix}, \tag{7.72}$$

where v_{n_1} was just taken as v_n.

7.7.3 A NOTE ON NUMERICAL METHODS AND COMPUTATIONS

Three different types of numerical methods were used in a combined manner: 1) multi-variable function root-finder, 2) ordinary differential equation (ODE) solver for initial value problems, and 3) centred finite-differences about mid-segments. Details about its usage are as follows:

- Root finder: all problems of the form $\mathrm{f}(\mathbf{x}) = \mathbf{0}$ were solved using Python's ScyPy optimisation module. The function used is a wrapper that uses FORTRAN-MINPACK's `hybrd` and `hybrj` algorithms [111].
- ODE solver: all initial value problems of the form $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} = \mathbf{F}(s, \mathbf{x})$, with $\mathbf{x}(0) = \mathbf{x}_0$, where solved using Python's ScyPy integration module. The function used solves real-valued variable-coefficient ordinary differential equations, with fixed-leading-coefficient implementation, using backward differentiation formulas (BDF) for stiff problems [110]. Given

that all problems are boundary-valued, not initial-valued, this algorithm was used along with the root-finder. Here, a shooting-algorithm-type scheme was used, where the root-finder calculates the initial conditions that meet the boundary conditions requirement.

- Finite-differences scheme: all partial-differential equations of the form

$$K(Y,s)\frac{\partial Y}{\partial s} + F(Y,s) = 0$$

where space-discretised using centered differences about half-grid points, as proposed in [53, 54]. This means that the derivative is discretised so that

$$\frac{\partial Y}{\partial s} \approx \frac{Y_j - Y_{j-1}}{\Delta s},$$

where Δs is the discretised cable segment size. For the other functions evaluation at mid-point is an average of adjacent grid-points, this means that, *e.g.*,

$$K(Y_{j-\frac{1}{2}}, s_{j-\frac{1}{2}}) \approx \frac{1}{2}\left(K(Y_{j-1}, s_{j-1}) + K(Y_j, s_j)\right).$$

This converts the differential equation into a problem of the form $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, that is solved using the root-finder previously discussed.

All cases simulate a cable that hangs between two points and are subject to the external and internal loads specified. The first (left) hang point is considered fixed at the origin ($\hat{x}_0 = 0$ and $\hat{z}_0 = 0$), the second (right) hang point is assumed fixed at a known $(\hat{x}_N, \hat{z}_N)$ position. The different solutions are parametrised using the cable end-point coordinates in polar form; this means that variables $l = \sqrt{\hat{x}_N^2 + \hat{z}_N^2}$ and $\theta_l = \text{atan}(\hat{x}_N/\hat{z}_N)$ are used to organise the different computations. The following values are used to illustrate the results:

- $l = 0.1$, 0.3, 0.5, 0.7, 0.8, and 0.9.
- $\theta_l = 0$, 15, 30, 45, 60, 75, and 90 deg.

When drag is taken into account a constant current profile given by $\hat{v}_x = -1$ and $\hat{v}_z = 0$ is assumed. It is also assumed that drag acts directly on the cable's normal direction and is negligible on the tangential direction, this means that $f_n(\varphi) = 1$ and $f_t(\varphi) = 0$.

7.7.4 CASE 1: INEXTENSIBLE CABLE SUBJECT TO WEIGHT

For an inextensible cable with infinite bending flexibility subject only to weight loads the following simplifications can be performed:

- The internal load is only tangential, meaning that $\mathbf{F} = \begin{bmatrix} 0 & 0 & T \end{bmatrix}^{\mathsf{T}}$.

- Strain is zero, *i.e.*, $\epsilon = 0$.

Consequently, Eq. (7.59) becomes

$$\frac{\partial}{\partial s}\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\partial \theta}{\partial s} \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} -\bar{w}\sin\theta \\ 0 \\ \bar{w}\cos\theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

which leads to the system of equations

$$\frac{\partial T}{\partial s} + \bar{w}\cos\theta = 0 \quad \text{and} \tag{7.73a}$$

$$T\frac{\partial \theta}{\partial s} - \bar{w}\sin\theta = 0. \tag{7.73b}$$

These equations in non-dimensional form become

$$\frac{\partial \hat{T}}{\partial \hat{s}} + \cos\theta = 0 \quad \text{and} \tag{7.74a}$$

$$\hat{T}\frac{\partial \theta}{\partial \hat{s}} - \sin\theta = 0. \tag{7.74b}$$

This model has closed analytical solution given that initial conditions $\hat{T}(0) = \hat{T}_0$, $\theta(0) = \theta_0$, $\hat{x}(0) = 0$, and $\hat{z}(0) = 0$ are known. The solution [64], assuming that $h = \hat{T}_0 \sin\theta_0$ and $v = \hat{T}_0 \cos\theta_0$, is

$$\begin{aligned} \hat{T} =& \sqrt{h^2 + (v - \hat{s})^2} \\ \hat{x} =& h\left(\operatorname{asinh}\left(\frac{v}{h}\right) - \operatorname{asinh}\left(\frac{v - \hat{s}}{h}\right)\right), \quad \text{and} \\ \hat{z} =& \sqrt{h^2 + v^2} - \sqrt{h^2 - (v - \hat{s})^2}. \end{aligned}$$

Since usually h and v are not known beforehand and the boundary conditions $\hat{x}(1) = \hat{x}_N$ and $\hat{z}(1) = z_N$ are given, it is possible to configure this problem as a system of non-linear algebraic equations that is solvable using a root-finder. After solving the equations, the resulting cable profiles are shown in Fig. 7.12. Here, different colours represent different θ_l.

The behaviour of tension is shown as a function of l in Fig. 7.13 and as a function of θ_l in Fig. 7.14. Continuous lines represent tension at the end-point; dashed lines represent tension at the start-point. These results show that for small span between hanging points l and for values of θ_l approaching 90 deg initial and end tensions tend to be similar, they differ when l approaches full cable length and when θ_l approaches 0. It is important to consider that values of tension are referenced here to the complete cable weight, this means that a tension equal to 1 is equal to the total cable weight. Lower values of tension are generated when the profile is close to vertical and l is at least lower than 0.8.

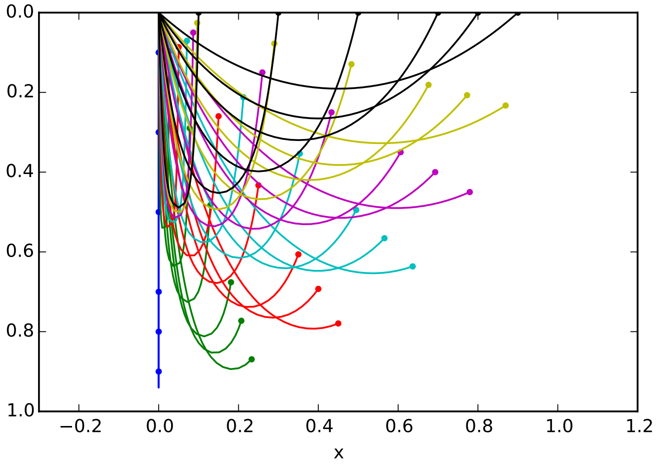


Figure 7.12. Cable profile—case 1

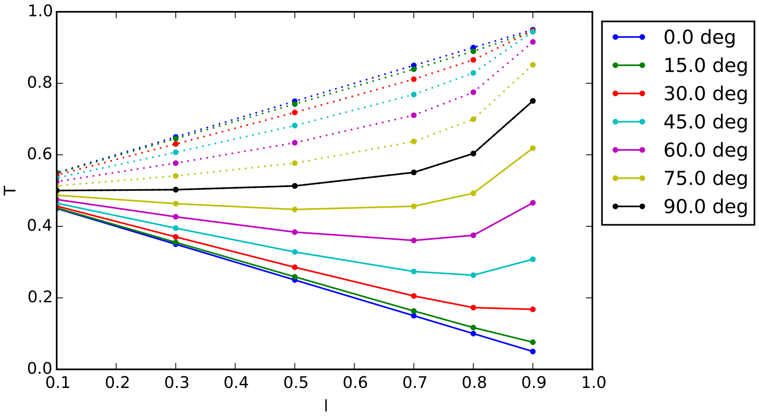


Figure 7.13. Cable tension as a function of l—case 1

7.7.5 CASE 2: EXTENSIBLE CABLE SUBJECT TO WEIGHT

For an extensible cable with infinite bending flexibility subject only to weight loads, as in the previous case, the internal load is only tangential, meaning that $\mathbf{F} = \begin{bmatrix} 0 & 0 & T \end{bmatrix}^{\mathsf{T}}$ still holds. For this case, Eq. (7.59) becomes

$$\frac{\partial}{\partial s}\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\partial \theta}{\partial s} \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} -\bar{w}\sin\theta \\ 0 \\ \bar{w}\cos\theta \end{bmatrix}(1+\epsilon) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

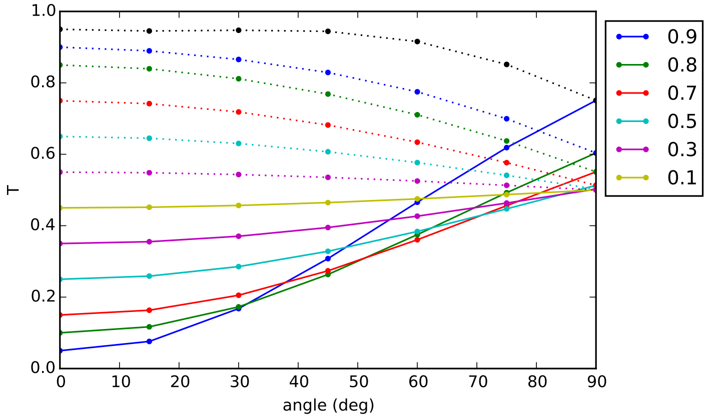


Figure 7.14. Cable tension as a function of θ_l—case 1

which leads to the system of equations

$$\frac{\partial T}{\partial s} + \bar{w}\cos\theta(1+\epsilon) = 0 \quad \text{and} \tag{7.76a}$$

$$T\frac{\partial \theta}{\partial s} - \bar{w}\sin\theta(1+\epsilon) = 0. \tag{7.76b}$$

These equations in non-dimensional form become

$$\frac{\partial \hat{T}}{\partial \hat{s}} + \cos\theta(1+c_s\hat{T}) = 0 \quad \text{and} \tag{7.77a}$$

$$\hat{T}\frac{\partial \theta}{\partial \hat{s}} - \sin\theta(1+c_s\hat{T}) = 0. \tag{7.77b}$$

This model has closed analytical solution given that initial conditions $\hat{T}(0) = \hat{T}_0$, $\theta(0) = \theta_0$, $\hat{x}(0) = 0$, and $z(0) = 0$ are known. The solution [64], assuming that $h = \hat{T}_0 \sin\theta_0$ and $v = \hat{T}_0\cos\theta_0$, is

$$\begin{aligned}
\hat{T} =& \sqrt{h^2 + (v-\hat{s})^2} \\
\hat{x} =& h\left(c_s\hat{s} + \operatorname{asinh}\left(\frac{v}{h}\right) - \operatorname{asinh}\left(\frac{v-\hat{s}}{h}\right)\right), \quad \text{and} \\
\hat{z} =& c_s\hat{s}\left(v - \frac{\hat{s}}{2}\right) + \sqrt{h^2+v^2} - \sqrt{h^2-(v-\hat{s})^2}.
\end{aligned}$$

As in the previous case, this problem can be written as a system of non-linear algebraic equations that is solvable using a root-finder; also the problem is parametrised similarly as well. The only difference between this and the previous case lies in the fact that in this case different values of c_s are considered, namely $c_s = 0$, 0.001, 0.01, and 0.1. After solving the equations, the resulting cable profiles are shown in Fig. 7.15. Here, the slight variation of

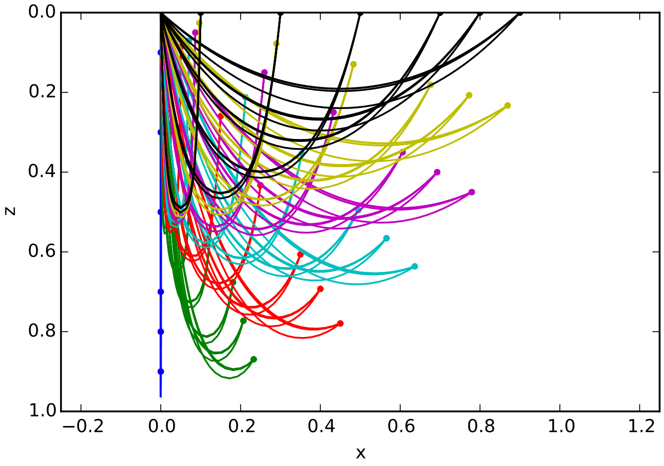


Figure 7.15. Cable profile—case 2

the profile at each position is given by the variation of c_s, where the most sag is found when $c_s = 0.1$.

The behaviour of tension is shown as a function of l in Fig. 7.16 and as a function of θ_l in Fig. 7.17. Continuous lines represent tension at the end-point; dashed lines represent tension at the start-point. It is apparent from the results that flexibility of the cable becomes important when the cable approaches its full extension; here, tension differs among different values of c_s when $l > 0.8$.

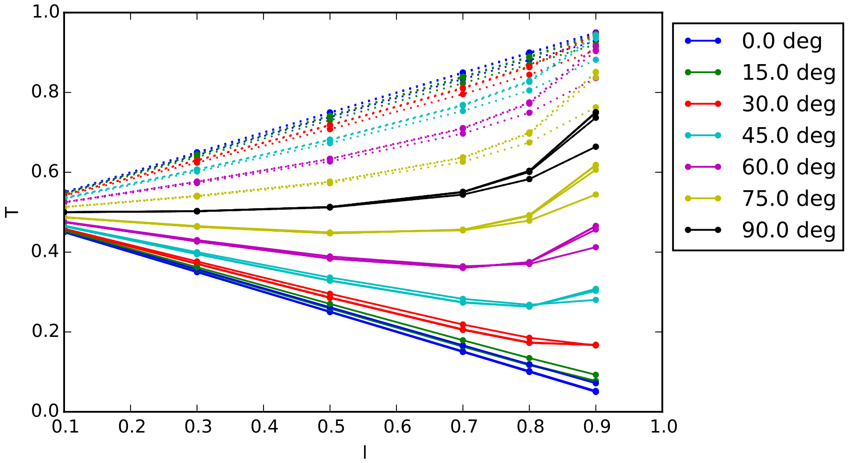


Figure 7.16. Cable tension as a function of l—case 2

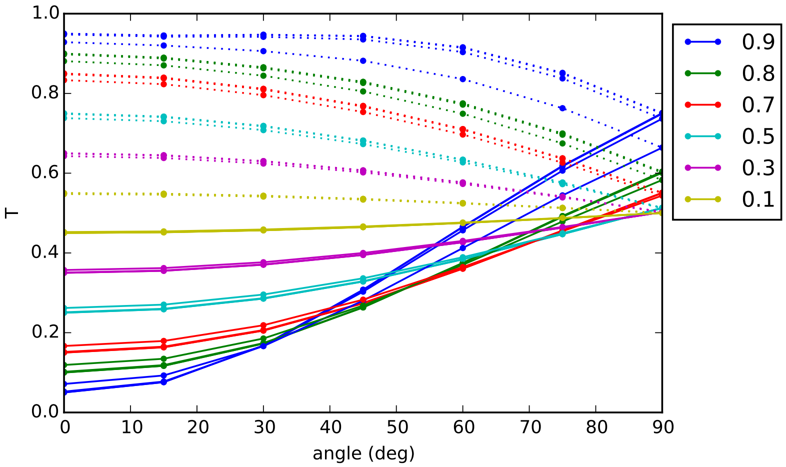


Figure 7.17. Cable tension as a function of θ_l—case 2

7.7.6 CASE 3: INEXTENSIBLE CABLE SUBJECT TO WEIGHT AND DRAG

Relating simplifications, this case is similar to case 1, just that in this case the drag load is included. Consequently, for this case Eq. (7.59) becomes

$$\frac{\partial}{\partial s}\begin{bmatrix}0\\0\\T\end{bmatrix}+\begin{bmatrix}0\\\frac{\partial\theta}{\partial s}\\0\end{bmatrix}\times\begin{bmatrix}0\\0\\T\end{bmatrix}+\begin{bmatrix}-\bar{w}\sin\theta\\0\\\bar{w}\cos\theta\end{bmatrix}+\begin{bmatrix}-\frac{1}{2}\rho_{\mathrm{w}}dC_d f_n(\varphi)v_n|v_n|\\0\\-\frac{1}{2}\rho_{\mathrm{w}}dC_d f_t(\varphi)v_t|v_t|\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix},$$

which leads to the system of equations

$$\frac{\partial T}{\partial s}+\bar{w}\cos\theta-\frac{1}{2}\rho_{\mathrm{w}}dC_d f_t(\varphi)v_t|v_t|=0 \quad \text{and} \tag{7.79a}$$

$$T\frac{\partial\theta}{\partial s}-\bar{w}\sin\theta-\frac{1}{2}\rho_{\mathrm{w}}dC_d f_n(\varphi)v_n|v_n|=0. \tag{7.79b}$$

These equations in non-dimensional form become

$$\frac{\partial\hat{T}}{\partial\hat{s}}+(1-\hat{d})\cos\theta-\hat{d}f_t(\varphi)\hat{v}_t|\hat{v}_t|=0 \quad \text{and} \tag{7.80a}$$

$$\hat{T}\frac{\partial\theta}{\partial\hat{s}}-(1-\hat{d})\sin\theta-\hat{d}f_n(\varphi)\hat{v}_n|\hat{v}_n|=0. \tag{7.80b}$$

This model can be solved by either using a shooting algorithm or finite differences. In addition to the previously discussed parametrisation, cases involving drag are parametrised using $\hat{d}$. The following values were tested $\hat{d}$ = 0.1, 0.3, 0.5, 0.7, and 0.9. The resulting cable profiles, after computing the models, are shown in Fig. 7.18. If the results of this figure are compared to those on Fig. 7.12 (or even Fig. 7.15), the effect of drag becomes apparent at shaping the profile of the cable.

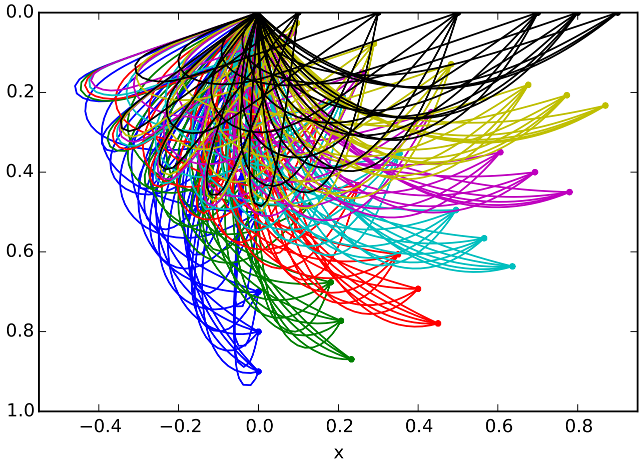


Figure 7.18. Cable profile—case 3

The behaviour of tension is shown as a function of l in Fig. 7.19 and as a function of θ_l in Fig. 7.20. The different colours represent different values of $\hat{d}$; for Fig. 7.19 at each colour several values of θ_l are represented and for Fig. 7.20 at each colour several values of l are represented. From these results, it is apparent that for higher values of $\hat{d}$ (*i.e.*, 0.7 or 0.9) tension is higher when l approaches 1 and angle is closer to 0, this is due to larger amount of cable facing drag. In the case of lower l, tension is less dependent on θ_l and mostly dependent on $\hat{d}$ only.

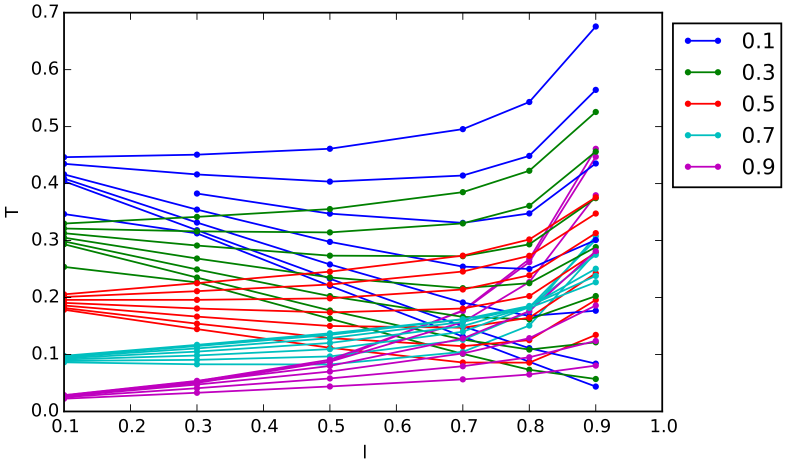


Figure 7.19. Cable tension as a function of l—case 3

The behaviour of tension as a function of $\hat{d}$ is shown in Fig. 7.21; here the different colours

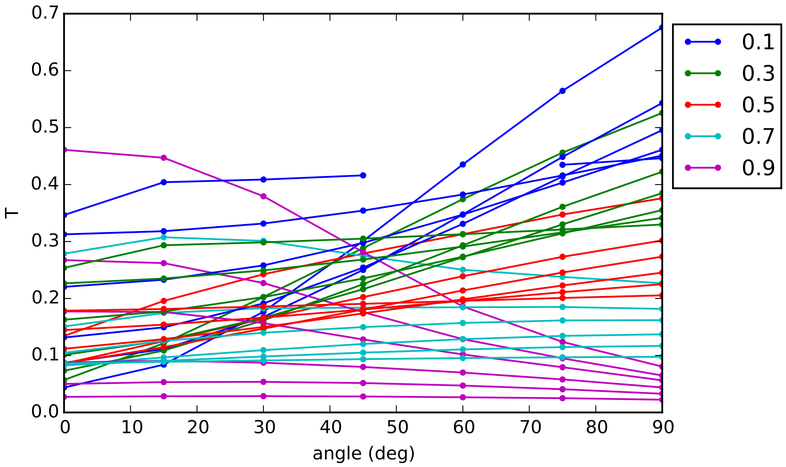


Figure 7.20. Cable tension as a function of θ_l—case 3

represent different values of l. From this figure it is interesting to see what happens when $\hat{d}$ ranges between 0.5 and 0.7, where values of tension are approximately independent on l or θ and relative lower than at other regions, at these values drag effect is similar or slightly higher than weight. It is interesting also to see that for $l < 0.8$ tension values are significantly low when drag increases, compared to those on different regions of the plot.

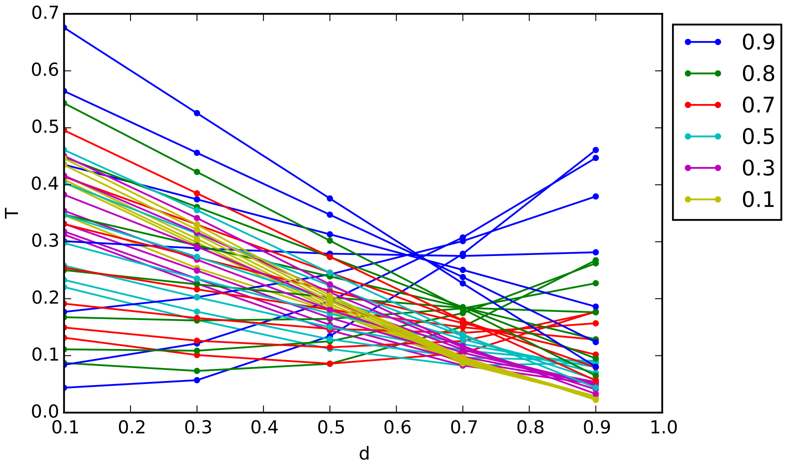


Figure 7.21. Cable tension as a function of $\hat{d}$—case 3

7.7.7 CASE 4: EXTENSIBLE CABLE SUBJECT TO WEIGHT AND DRAG

This case is similar to case 2, but in this case the drag load is included. For this case Eq. (7.59) becomes

$$\frac{\partial}{\partial s}\begin{bmatrix}0\\0\\T\end{bmatrix}+\begin{bmatrix}0\\\frac{\partial\theta}{\partial s}\\0\end{bmatrix}\times\begin{bmatrix}0\\0\\T\end{bmatrix}$$
$$+\left(\begin{bmatrix}-\bar{w}\sin\theta\\0\\\bar{w}\cos\theta\end{bmatrix}+\begin{bmatrix}-\frac{1}{2}\rho_{\mathrm{w}}dC_df_n(\varphi)v_n|v_n|\\0\\-\frac{1}{2}\rho_{\mathrm{w}}dC_df_t(\varphi)v_t|v_t|\end{bmatrix}\right)(1+\epsilon)=\begin{bmatrix}0\\0\\0\end{bmatrix},$$

which leads to the system of equations

$$\frac{\partial T}{\partial s}+\left(\bar{w}\cos\theta-\frac{1}{2}\rho_{\mathrm{w}}dC_df_n(\varphi)v_n|v_n|\right)(1+\epsilon)=0\quad\text{and}\tag{7.81a}$$
$$T\frac{\partial\theta}{\partial s}+\left(-\bar{w}\sin\theta-\frac{1}{2}\rho_{\mathrm{w}}dC_df_t(\varphi)v_t|v_t|\right)(1+\epsilon)=0.\tag{7.81b}$$

These equations in non-dimensional form become

$$\frac{\partial\hat{T}}{\partial\hat{s}}+((1-\hat{d})\cos\theta-\hat{d}f_t(\varphi)\hat{v}_t|\hat{v}_t|)(1+c_s\hat{T})=0\quad\text{and}\tag{7.82a}$$
$$\hat{T}\frac{\partial\theta}{\partial\hat{s}}+(-(1-\hat{d})\sin\theta-\hat{d}f_n(\varphi)\hat{v}_n|\hat{v}_n|)(1+c_s\hat{T})=0.\tag{7.82b}$$

This model is solved similarly as in the previous case, regarding numerical methods and parametrisation, but also in this case c_s is parametrised as in case 3, *i.e.*, $c_s = 0$, 0.001, 0.01, and 0.1. Comparing these results to those on Fig. 7.18 it is apparent that the cable profile is only slightly modified when the cable becomes less rigid. The resulting cable profiles, after computing the models, are shown in Fig. 7.22.

The behaviour of tension is shown as a function of l in Fig. 7.23 and as a function of θ_l in Fig. 7.24. As in the previous case, the different colours represent different values of $\hat{d}$; for Fig. 7.23 at each colour several values of θ_l are represented and for Fig. 7.24 at each colour several values of l are represented.

The behaviour of tension as a function of $\hat{d}$ is shown in Fig. 7.25; here the different colours represent different values of l. Most of the conclusions on the previous case hold at this one, as it become apparent from the figures, tension is only slightly modified when the cable becomes less rigid.

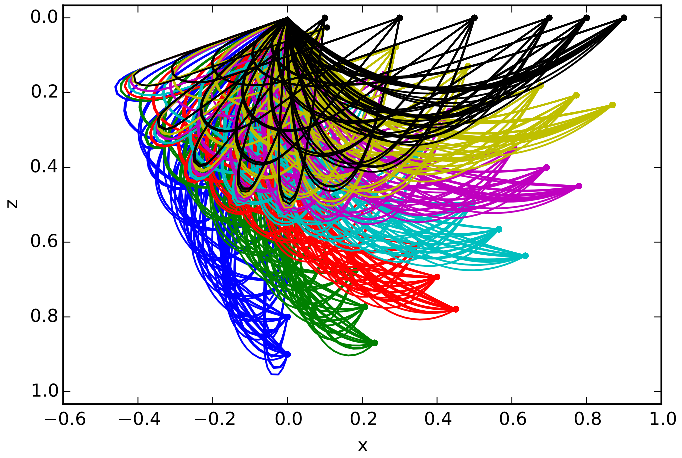


Figure 7.22. Cable profile—case 4

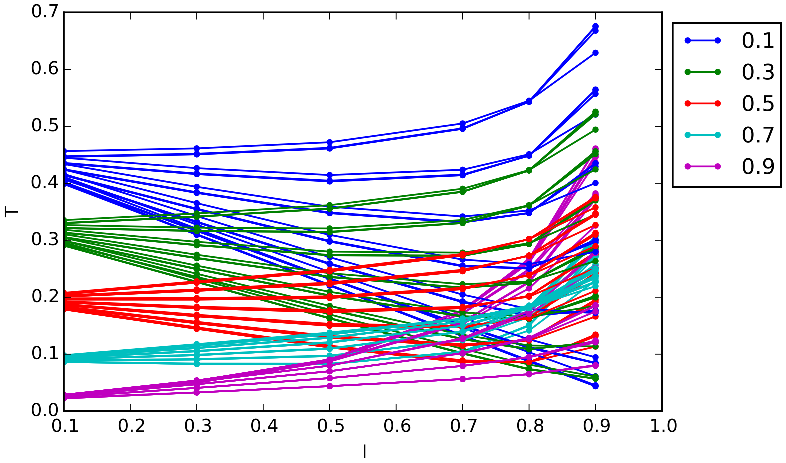


Figure 7.23. Cable tension as a function of l—case 4

7.7.8 CASE 5: INEXTENSIBLE CABLE SUBJECT TO WEIGHT AND DRAG, INCLUDING BENDING EFFECTS

When bending is considered internal loads are not only due to axial strain, but for shear strain as well. Here the moments equation in (7.61) becomes relevant, which for the two-dimensional case can be rewritten considering that $\mathbf{M} = \begin{bmatrix} 0 & EI\Omega & 0 \end{bmatrix}^\intercal$ and $\mathbf{F} = \begin{bmatrix} F_n & 0 & T \end{bmatrix}^\intercal$. This

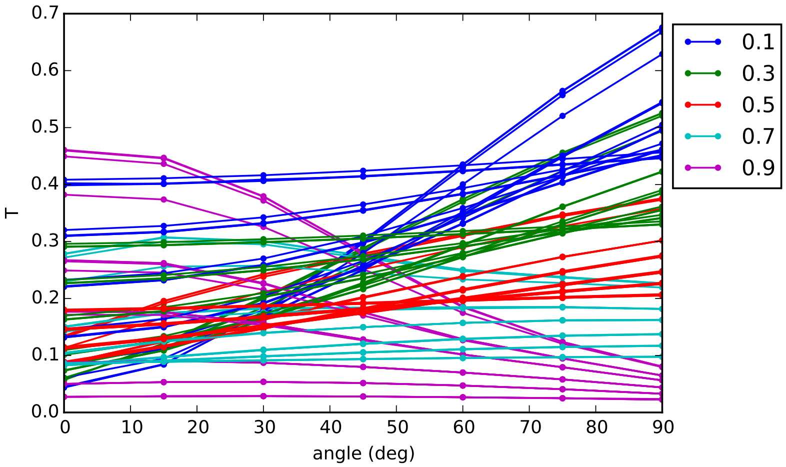


Figure 7.24. Cable tension as a function of θ_l—case 4

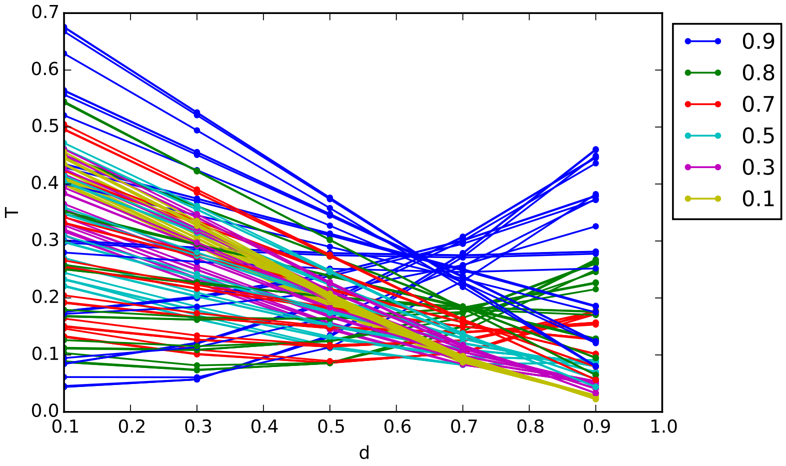


Figure 7.25. Cable tension as a function of $\hat{d}$—case 4

becomes

$$\frac{\partial}{\partial s}\begin{bmatrix}0\\EI\Omega\\0\end{bmatrix}+\begin{bmatrix}0\\\Omega\\0\end{bmatrix}\times\begin{bmatrix}0\\EI\Omega\\0\end{bmatrix}+\begin{bmatrix}0\\0\\1\end{bmatrix}\times\begin{bmatrix}F_n\\0\\T\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix},$$

where the expression of the shear force is found as

$$S_n=-EI\frac{\partial\Omega}{\partial s}.$$

This redefines the internal force as $\mathbf{F} = \begin{bmatrix} S_n & 0 & T \end{bmatrix}^\mathsf{T}$. Then, the forces equation (7.59) becomes

$$\frac{\partial}{\partial s}\begin{bmatrix} S_n \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ \Omega \\ 0 \end{bmatrix} \times \begin{bmatrix} S_n \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} -\bar{w}\sin\theta \\ 0 \\ \bar{w}\cos\theta \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}\rho_\mathrm{w} dC_d f_n(\varphi) v_n |v_n| \\ 0 \\ -\frac{1}{2}\rho_\mathrm{w} dC_d f_t(\varphi) v_t |v_t| \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

which leads to the system of equations

$$\frac{\partial T}{\partial s} - S_n\Omega + \bar{w}\cos\theta - \frac{1}{2}\rho_\mathrm{w} dC_d f_n(\varphi) v_n|v_n| = 0 \tag{7.83a}$$

$$\frac{\partial S_n}{\partial s} + T\Omega - \bar{w}\sin\theta - \frac{1}{2}\rho_\mathrm{w} dC_d f_t(\varphi) v_t|v_t| = 0, \quad \text{and} \tag{7.83b}$$

$$EI\frac{\partial \Omega}{\partial s} + S_n = 0. \tag{7.83c}$$

These equations in non-dimensional form become

$$\frac{\partial \hat{T}}{\partial \hat{s}} - \hat{S}_n\hat{\Omega} + (1-\hat{d})\cos\theta - \hat{d} f_n(\varphi)\hat{v}_n|\hat{v}_n| = 0 \tag{7.84a}$$

$$\frac{\partial \hat{S}_n}{\partial \hat{s}} + \hat{T}\hat{\Omega} - (1-\hat{d})\sin\theta - \hat{d} f_t(\varphi)\hat{v}_t|\hat{v}_t| = 0, \quad \text{and} \tag{7.84b}$$

$$\frac{1}{c_b}\frac{\partial \hat{\Omega}}{\partial \hat{s}} + \hat{S}_n = 0. \tag{7.84c}$$

The model is solved using finite differences only; it was difficult to use a shooting algorithm to solve this model: when guessing different initial conditions the root-finding algorithm failed to match the stated boundary conditions, giving way to non-realistic cable geometries. These computations where parametrised also using different values of c_b, *i.e.*, c_b = 100, 1000, 10000, and 100000; greater values of c_b represent negligible bending stiffness.

The resulting cable profiles, after computing the models, are shown in Fig. 7.26. Comparing these results to those on Fig. 7.18 and 7.22 it is apparent that including bending effects affect the way the profile is distributed in space; a greater bending stiffness (lower c_b) generates a greater bending radius. Also, it can be observed that greater bending stiffness (lower c_b) reduces sag below the left hanging point.

The behaviour of tension is shown as a function of l in Fig. 7.27 and as a function of θ_l in Fig. 7.28. As in previous cases, the different colours represent different values of $\hat{d}$; for Fig. 7.27 at each colour several values of θ_l are represented and for Fig. 7.28 at each colour several values of l are represented. Also, given that c_b was varied as well, the effect of c_b variation appears in the plots as dispersion of the lines (different lines plotted next to each other). The behaviour of tension as a function of $\hat{d}$ is shown in Fig. 7.29; here the different colours represent different values of l.

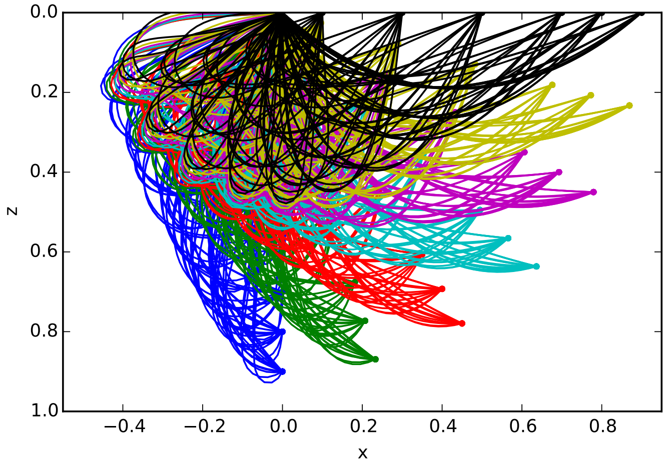


Figure 7.26. Cable profile—case 5

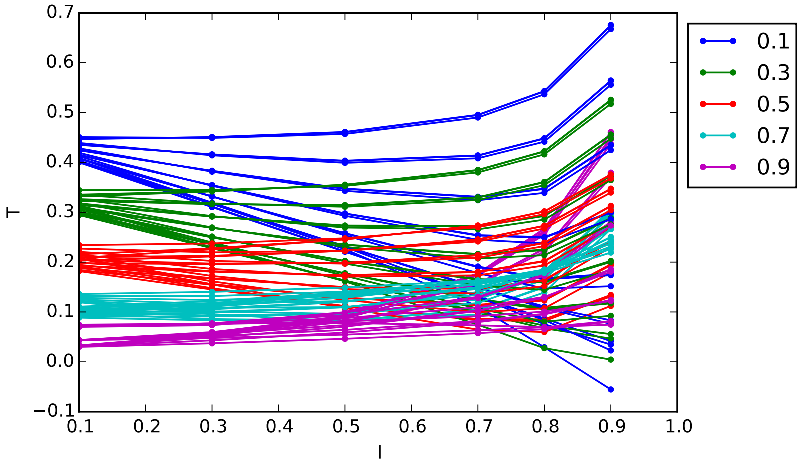


Figure 7.27. Cable tension as a function of l—case 5

The results on these plots suggest that bending stiffness does not greatly affect the value of tension at the end-point. Values of tension are still greatly modified by $\hat{d}$, which shows the relative importance of drag compared to weight, and by l, how is the span between hanging points compared to the total cable length.

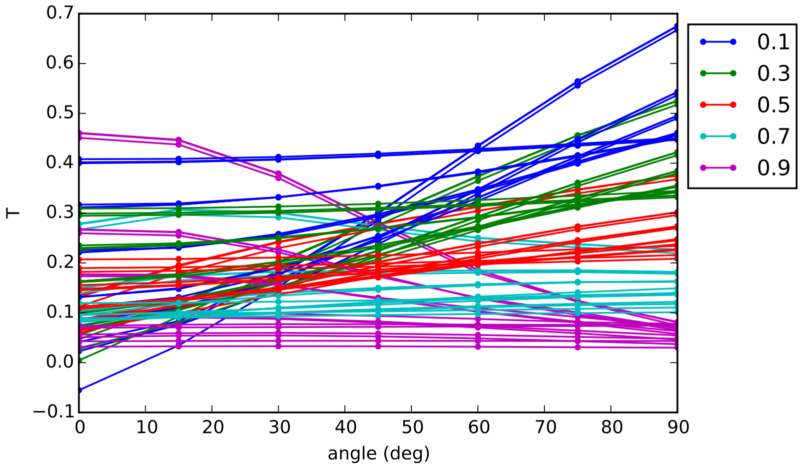


Figure 7.28. Cable tension as a function of θ_l—case 5

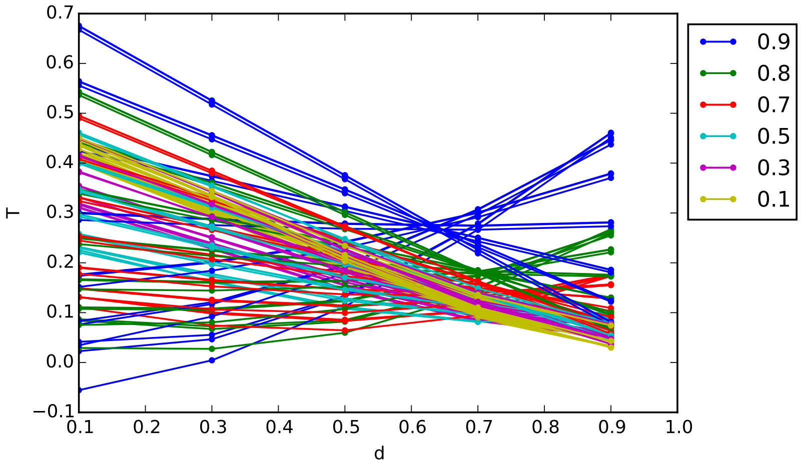


Figure 7.29. Cable tension as a function of $\hat{d}$—case 5

7.7.9 CASE 6: EXTENSIBLE CABLE SUBJECT TO WEIGHT AND DRAG, INCLUDING BENDING EFFECTS

Following a similar procedure as in the previous case, but considering strain, the moments equation becomes

$$\frac{\partial}{\partial s}\begin{bmatrix}0\\EI\Omega\\0\end{bmatrix}+\begin{bmatrix}0\\\Omega\\0\end{bmatrix}\times\begin{bmatrix}0\\EI\Omega\\0\end{bmatrix}+\begin{bmatrix}0\\0\\1\end{bmatrix}\times\begin{bmatrix}F_n\\0\\T\end{bmatrix}(1+\epsilon)=\begin{bmatrix}0\\0\\0\end{bmatrix},$$

from which the obtained expression of the shear force is

$$S_n = -\frac{EI}{(1+\epsilon)}\frac{\partial\Omega}{\partial s}.$$

This redefines the internal force as

$$\mathbf{F} = \begin{bmatrix} S_n & 0 & T \end{bmatrix}^{\mathsf{T}}.$$

Then, the forces equation (7.59) becomes

$$\frac{\partial}{\partial s}\begin{bmatrix} S_n \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ \Omega \\ 0 \end{bmatrix} \times \begin{bmatrix} S_n \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} -\bar{w}\sin\theta \\ 0 \\ \bar{w}\cos\theta \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}\rho_{\mathrm{w}} d C_d f_n(\varphi) v_n |v_n| \\ 0 \\ -\frac{1}{2}\rho_{\mathrm{w}} d C_d f_t(\varphi) v_t |v_t| \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

which leads to the system of equations

$$\frac{\partial T}{\partial s} - S_n\Omega + \left(\bar{w}\cos\theta - \frac{1}{2}\rho_{\mathrm{w}} d C_d f_n(\varphi) v_n |v_n|\right)(1+\epsilon) = 0, \tag{7.85a}$$

$$\frac{\partial S_n}{\partial s} + T\Omega + \left(-\bar{w}\sin\theta - \frac{1}{2}\rho_{\mathrm{w}} d C_d f_t(\varphi) v_t |v_t|\right)(1+\epsilon) = 0, \quad \text{and} \tag{7.85b}$$

$$EI\frac{\partial\Omega}{\partial s} + S_n(1+\epsilon) = 0. \tag{7.85c}$$

These equations in non-dimensional form become

$$\frac{\partial\hat{T}}{\partial\hat{s}} - \hat{S}_n\hat{\Omega} + \left((1-\hat{d})\cos\theta - \hat{d} f_n(\varphi)\hat{v}_n|\hat{v}_n|\right)(1+c_s\hat{T}) = 0, \tag{7.86a}$$

$$\frac{\partial\hat{S}_n}{\partial\hat{s}} + \hat{T}\hat{\Omega} + \left(-(1-\hat{d})\sin\theta - \hat{d} f_t(\varphi)\hat{v}_t|\hat{v}_t|\right)(1+c_s\hat{T}) = 0, \quad \text{and} \tag{7.86b}$$

$$\frac{1}{c_b}\frac{\partial\hat{\Omega}}{\partial\hat{s}} + \hat{S}_n(1+c_s\hat{T}) = 0. \tag{7.86c}$$

This model is solved mostly as in the previous case regarding numerical methods and parametrisation, but in this case c_s is parametrised as $c_s = 0$, 0.001, 0.01, and 0.1 and all computations use a unique value for bending flexibility $c_b = 1000$.

The resulting cable profiles, after computing the models, are shown in Fig. 7.30. As in other previous results, comparing these results to those on Fig. 7.26 it is apparent that the cable profile is only slightly modified when the cable becomes less rigid.

The behaviour of tension is shown as a function of l in Fig. 7.31 and as a function of θ_l in Fig. 7.32. As in previous cases, the different colours represent different values of $\hat{d}$; for Fig. 7.31 at each colour several values of θ_l are represented and for Fig. 7.32 at each colour several values of l are represented. Also, given that c_s was varied as well, the effect of c_c variation appears in the plots as dispersion of the lines. The behaviour of tension as a function of $\hat{d}$ is shown in Fig. 7.33; here the different colours represent different values of l. Most of the conclusions on the previous case hold at this one, as it becomes apparent from the figures, tension is only slightly modified when the cable becomes less rigid.

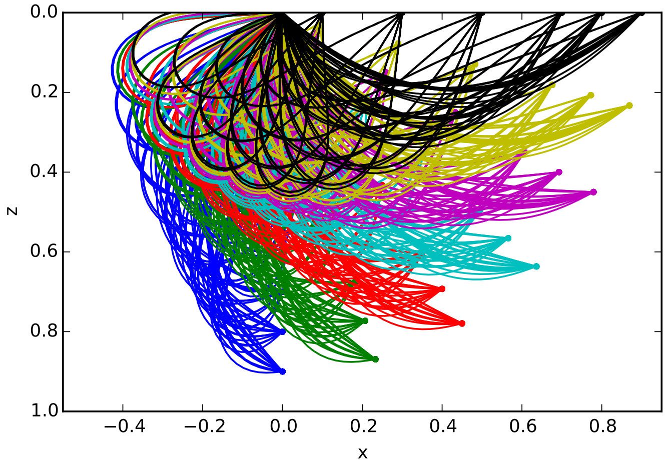


Figure 7.30. Cable profile—case 6

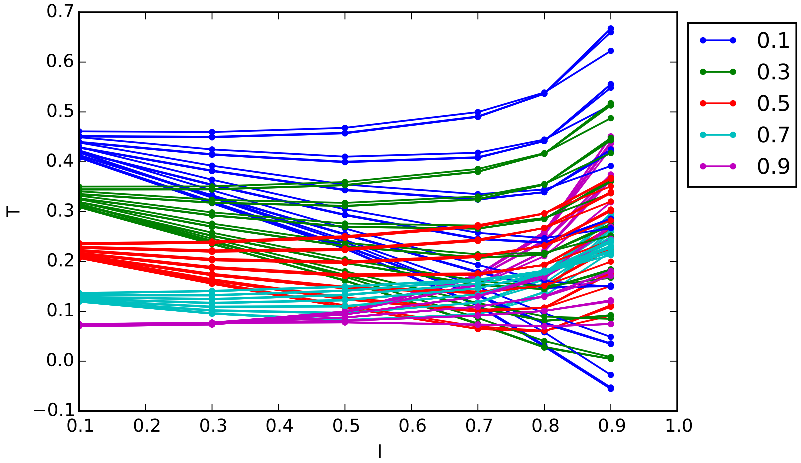


Figure 7.31. Cable tension as a function of l—case 6

7.8 DYNAMICS COMPUTATIONS USING CABLE COORDINATES

In this study the 2-dimensional dynamics analysis of an underwater cable is taken into account. During the analysis the following phenomena are considered:

- Drag due to current.
- Weight and buoyancy.
- Rigid body mass and hydrodynamic added mass.

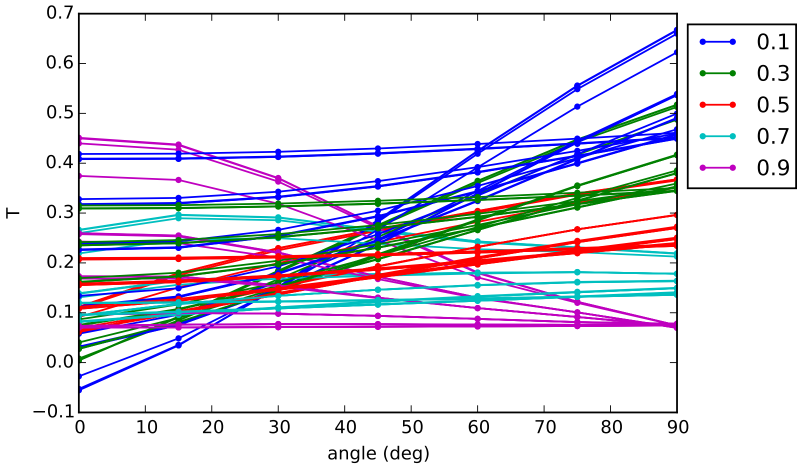


Figure 7.32. Cable tension as a function of θ_l—case 6

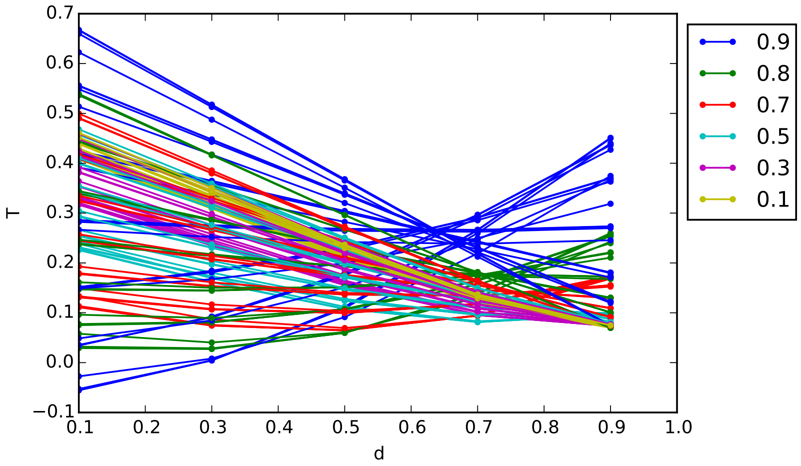


Figure 7.33. Cable tension as a function of $\hat{d}$—case 6

- Tension and bending stiffness.

Throughout this section, the ideas regarding modelling and numerical schemes on [62, 53, 54] are followed; nevertheless, the model specifics are included in this document.

7.8.1 NON-DIMENSIONAL VARIABLES

Calculation of non-dimensional variables is based on three basic scaling quantities: the cable's non-stretched length L, mass per unit length of the cable m, and gravity g. Application of these quantities gives the following non-dimensional variables:

- Time is scaled as $\hat{t} = \sqrt{\frac{g}{L}}t$
- Distance-based variables are scaled as $\hat{s} = s/L$, $\hat{x} = x/L$, $\hat{y} = y/L$, and $\hat{z} = z/L$.
- Curvature-related variables are scaled, for $i = 1, 2,$ and 3, as $\hat{\Omega}_i = L\Omega_i$, $\frac{\mathrm{d}\hat{\Omega}_i}{\mathrm{d}\hat{s}} = L^2\frac{\mathrm{d}\Omega_i}{\mathrm{d}s}$, and $\frac{\mathrm{d}^2\hat{\Omega}_i}{\mathrm{d}\hat{s}^2} = L^3\frac{\mathrm{d}^2\Omega_i}{\mathrm{d}s^2}$.
- Angular velocity variables are scaled, for $i = 1, 2,$ and 3, as $\hat{\omega}_i = \sqrt{\frac{L}{g}}\omega_i$.
- Linear velocity variables are scaled as $\hat{u} = \sqrt{\frac{1}{gL}}u$, $\hat{v} = \sqrt{\frac{1}{gL}}v$, and $\hat{w} = \sqrt{\frac{1}{gL}}w$.
- Linear acceleration variables are scaled as $\frac{\partial\hat{u}}{\partial t} = \frac{1}{g}\frac{\partial u}{\partial t}$, $\frac{\partial\hat{v}}{\partial t} = \frac{1}{g}\frac{\partial v}{\partial t}$, and $\frac{\partial\hat{w}}{\partial t} = \frac{1}{g}\frac{\partial w}{\partial t}$.
- Tension is scaled as $\hat{T} = \frac{T}{mgL}$.
- Axial and bending flexibility are respectively scaled as $c_s = \frac{mgL}{EA}$ and $c_b = \frac{mgL^3}{EI}$. Following this, strain may be defined as function of tension as $\epsilon = c_s\hat{T}$.

7.8.2 DYNAMICS MODEL IN TWO DIMENSIONS

In the two-dimensional case the following simplifications are applied:

- Displacement, velocity and acceleration are restricted to the $xz-$plane.
- In the transformation (7.3), the rotation around the $x-$axis is eliminated, this means that $\phi = 0$.
- Drag loads, *i.e.*, current profiles, are restricted to the $xz-$plane.

2D assumptions are the same as those of the statics analyses. The equations for cable element's coordinates are given by(7.4)–(7.6)

$$\frac{\partial x}{\partial s} = \sin\theta(1+\epsilon),$$
$$\frac{\partial y}{\partial s} = 0, \quad \text{and}$$
$$\frac{\partial z}{\partial s} = \cos\theta(1+\epsilon).$$

The curvatures vector is given by (7.70)

$$\mathbf{\Omega} = \begin{bmatrix} 0 \\ \Omega \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial\theta}{\partial s} \\ 0 \end{bmatrix}.$$

Angular velocities vector becomes

$$\omega = \begin{bmatrix} 0 \\ \omega \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial\theta}{\partial t} \\ 0 \end{bmatrix}.$$

The inertia term becomes

$$\mathbf{m}\left(\frac{\partial \mathbf{V}}{\partial t}+\omega\times\mathbf{V}\right)=\begin{bmatrix}(m+m_a)\left(\frac{\partial\hat{u}}{\partial t}+\hat{w}\frac{\partial\theta}{\partial t}\right)\\ 0\\ m\left(\frac{\partial\hat{w}}{\partial t}-\hat{u}\frac{\partial\theta}{\partial t}\right)\end{bmatrix}. \tag{7.88}$$

The compatibility equations (7.65a)–(7.65c) are reduced to

$$(1+\epsilon)\frac{\partial\theta}{\partial t}=\frac{\partial u}{\partial s}+w\frac{\partial\theta}{\partial s}, \tag{7.89a}$$

$$\frac{\partial\epsilon}{\partial t}=\frac{\partial w}{\partial s}-u\frac{\partial\theta}{\partial s}. \tag{7.89b}$$

The weight vector is given by (7.71)

$$\mathbf{w}=\begin{bmatrix}-(m-\rho_{\mathrm{w}}A)g\sin\theta\\ 0\\ (m-\rho_{\mathrm{w}}A)g\cos\theta\end{bmatrix}.$$

The drag vector is given by (7.72)

$$\mathbf{d}=-\frac{1}{2}\rho_{\mathrm{w}}dC_d\begin{bmatrix}f_n(\varphi)v_n|v_n|\\ 0\\ f_t(\varphi)v_t|v_t|\end{bmatrix},$$

where v_{n_1} was just taken as v_n.

In the 2-dimensional case the moments equation becomes

$$\frac{\partial}{\partial s}\begin{bmatrix}0\\ EI\Omega\\ 0\end{bmatrix}+\begin{bmatrix}0\\ \Omega\\ 0\end{bmatrix}\times\begin{bmatrix}0\\ EI\Omega\\ 0\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}\times\begin{bmatrix}S_n\\ 0\\ T\end{bmatrix}(1+\epsilon)=\begin{bmatrix}0\\ 0\\ 0\end{bmatrix},$$

from which the obtained expression of the shear force is

$$EI\frac{\partial\Omega}{\partial s}+S_n(1+\epsilon)=0. \tag{7.90}$$

This redefines the internal force as

$$\mathbf{F}=\begin{bmatrix}S_n & 0 & T\end{bmatrix}^T.$$

Then, the forces equation (7.64) becomes

$$\begin{bmatrix}(m+m_a)\left(\frac{\partial\hat{u}}{\partial t}+\hat{w}\frac{\partial\theta}{\partial t}\right)\\ 0\\ m\left(\frac{\partial\hat{w}}{\partial t}-\hat{u}\frac{\partial\theta}{\partial t}\right)\end{bmatrix}=\frac{\partial}{\partial s}\begin{bmatrix}S_n\\ 0\\ T\end{bmatrix}+\begin{bmatrix}0\\ \Omega\\ 0\end{bmatrix}\times\begin{bmatrix}S_n\\ 0\\ T\end{bmatrix}$$
$$+\begin{bmatrix}-(m-\rho_{\mathrm{w}}A)g\sin\theta\\ 0\\ (m-\rho_{\mathrm{w}}A)g\cos\theta\end{bmatrix}+\begin{bmatrix}-\frac{1}{2}\rho_{\mathrm{w}}dC_df_n(\varphi)v_n|v_n|\\ 0\\ -\frac{1}{2}\rho_{\mathrm{w}}dC_df_t(\varphi)v_t|v_t|\end{bmatrix}.$$

This leads to the equations

$$\begin{aligned}
(m+m_a)\left(\frac{\partial u}{\partial t}+w\frac{\partial \theta}{\partial t}\right) =& \frac{\partial S_n}{\partial s}+\mathit{\Omega} T-(m-\rho_{\mathrm{w}}A)g\sin\theta(1+\epsilon)\\
&-\frac{1}{2}\rho_{\mathrm{w}}dC_d f_t(\varphi)v_t|v_t|(1+\epsilon) && \text{(7.91a)}\\
m\left(\frac{\partial w}{\partial t}-u\frac{\partial \theta}{\partial t}\right) =& \frac{\partial T}{\partial \hat{s}}-\Omega S_n+(m-\rho_{\mathrm{w}}A)g\cos\theta(1+\epsilon)\\
&-\frac{1}{2}\rho_{\mathrm{w}}dC_d f_n(\varphi)v_n|v_n|(1+\epsilon). && \text{(7.91b)}
\end{aligned}$$

These equations are complemented with (7.69a), (7.70), (7.69c), (7.90), (7.89a), and (7.89b):

$$\begin{aligned}
(1+\epsilon)\frac{\partial \theta}{\partial t} &= \frac{\partial u}{\partial s}+w\frac{\partial \theta}{\partial s}, && \text{(7.92a)}\\
\frac{\partial \epsilon}{\partial t} &= \frac{\partial w}{\partial s}-u\frac{\partial \theta}{\partial s}, && \text{(7.92b)}\\
EI\frac{\partial \Omega}{\partial s} &= -S_n(1+\epsilon), && \text{(7.92c)}\\
\frac{\partial \theta}{\partial s} &= \Omega, && \text{(7.92d)}\\
\frac{\partial x}{\partial s} &= \sin\theta(1+\epsilon), && \text{(7.92e)}\\
\frac{\partial z}{\partial s} &= \cos\theta(1+\epsilon). && \text{(7.92f)}
\end{aligned}$$

7.8.3 DYNAMICS MODEL IN NON-DIMENSIONAL FORM

To formulate the equations in non-dimensional forms, in addition to the previously defined quantities, the following variables are defined:

- $\vartheta = \frac{\rho_w A}{m}$.
- $c_a = \frac{m_a}{m}$.
- $\ell = \frac{\mathrm{d}L}{A} = \frac{4L}{\pi \mathrm{d}}$.

Then, the equations in non-dimensional form become

$$\begin{aligned}
(1+c_a)\left(\frac{\partial \hat{u}}{\partial \hat{t}}+\hat{w}\frac{\partial \theta}{\partial \hat{t}}\right) =& \frac{\partial \hat{S}_n}{\partial \hat{s}}+\hat{\Omega}\hat{T} \\
&-(1-\vartheta)\sin\theta(1+c_s\hat{T}) \\
&-\frac{1}{2}\vartheta C_d\ell f_t(\varphi)\hat{v}_t|\hat{v}_t|(1+c_s\hat{T}), && (7.93a)\\
\frac{\partial \hat{w}}{\partial \hat{t}}-\hat{u}\frac{\partial \theta}{\partial \hat{t}} =& \frac{\partial \hat{T}}{\partial s}-\hat{\Omega}\hat{S}_n \\
&+(1-\vartheta)\cos\theta(1+c_s\hat{T}) \\
&-\frac{1}{2}\vartheta C_d\ell f_n(\varphi)\hat{v}_n|\hat{v}_n|(1+c_s\hat{T}), && (7.93b)\\
(1+c_s\hat{T})\frac{\partial \theta}{\partial \hat{t}} =& \frac{\partial \hat{u}}{\partial \hat{s}}+\hat{w}\frac{\partial \theta}{\partial \hat{s}}, && (7.93c)\\
c_s\frac{\partial \hat{T}}{\partial \hat{t}} =& \frac{\partial \hat{w}}{\partial \hat{s}}-\hat{u}\frac{\partial \theta}{\partial \hat{s}}, && (7.93d)\\
\frac{1}{c_b}\frac{\partial \hat{\Omega}}{\partial \hat{s}} =& -\hat{S}_n(1+c_s\hat{T}), && (7.93e)\\
\frac{\partial \theta}{\partial \hat{s}} =& \hat{\Omega}, && (7.93f)\\
\frac{\partial \hat{x}}{\partial \hat{s}} =& \sin\theta(1+c_s\hat{T}), && (7.93g)\\
\frac{\partial \hat{z}}{\partial \hat{s}} =& \cos\theta(1+c_s\hat{T}). && (7.93h)
\end{aligned}$$

7.8.4 NUMERICAL PROBLEM FORMULATION

This section follows the methods on [53, 54]. The spatial discretisation is done using centered differences about mid-segments and time discretisation is done using the generalised α-method. The PDE system on (7.93a)–(7.93h) may be written in matrix form as

$$M(Y,s)\frac{\partial Y}{\partial t}+K(Y,s)\frac{\partial Y}{\partial s}+F(Y,s)=0, \tag{7.94}$$

with $Y = \begin{bmatrix} u & w & \theta & T & \Omega & S_n & x & z \end{bmatrix}^T$. This equation system is spatially discretised using centred differences about mid-segments, *i.e.*,

$$\left.\frac{\partial Y}{\partial s}\right|_{j-1/2} \approx \frac{Y_j - Y_{j-1}}{\Delta s},$$

where $j = 0, 1, \ldots, N$ indicates the node number; this meas that the cable has N segments and $N+1$ nodes. The other components of the discretised version are assumed at the mid-segment and calculated as an average of the surrounding nodes, this means that (7.94) for each mid-segment becomes

$$M_{j-1}\dot{Y}_{j-1}+M_j\dot{Y}_j+\left(\frac{K_{j-1}+K_j}{\Delta s}\right)(Y_j-Y_{j-1})+F_{j-1}+F_j=0, \tag{7.95}$$

where the overdot means differentiation with respect to time, $M_j = M(Y_j, s_j)$, $K_j = K(Y_j, s_j)$, and $F_j = F(Y_j, s_j)$. From (7.95) a system of $8N$ differential-algebraic equations may be formed.

For time discretisation temporal weighted averages of the variables are used and for time integration the generalised trapezoidal rule is used. If the time-weighted average is applied to (7.95) each equation becomes

$$\begin{aligned}(1-\alpha_M)M_{j-1}^{i}\dot{Y}_{j-1}^{i} + \alpha_M M_{j-1}^{i-1}\dot{Y}_{j-1}^{i-1} + (1-\alpha_M)M_j^i\dot{Y}_j^i + \alpha_M M_j^{i-1}\dot{Y}_j^{i-1}&\\ + (1-\alpha_K)\left(\frac{K_{j-1}^i + K_j^i}{\Delta s}\right)\left(Y_j^i - Y_{j-1}^i\right)&\\ + \alpha_K\left(\frac{K_{j-1}^{i-1} + K_j^{i-1}}{\Delta s}\right)\left(Y_j^{i-1} - Y_{j-1}^{i-1}\right)&\\ + (1-\alpha_K)F_{j-1}^i + \alpha_K F_{j-1}^{i-1} + (1-\alpha_K)F_j^i + \alpha_K F_j^{i-1} = 0.&\end{aligned} \tag{7.96}$$

Then time integration of each variable Y_j is done using the generalised trapezoidal rule

$$Y_j^i = Y_j^{i-1} + \Delta t\left((1-\gamma)\dot{Y}_j^{i-1} + \gamma\dot{Y}_j^i\right). \tag{7.97}$$

Once (7.96) and (7.97) are defined for each mid-segment the time-advance problem can be written as a root-finding problem. For instance, if the initial condition of each variable $Y_j^{i-1} = Y_j^0$ is known and the next-time variable is $Y_j^i = Y_j$ is unknown, Eq. (7.96) may be rewritten as

$$\begin{aligned}(1-\alpha_M)M_{j-1}\dot{Y}_{j-1} + \alpha_M M_{j-1}^0\dot{Y}_{j-1}^0 + (1-\alpha_M)M_j\dot{Y}_j + \alpha_M M_j^0\dot{Y}_j^0&\\ + (1-\alpha_K)\left(\frac{K_{j-1} + K_j}{\Delta s}\right)\left(Y_j - Y_{j-1}\right)&\\ + \alpha_K\left(\frac{K_{j-1}^0 + K_j^0}{\Delta s}\right)\left(Y_j^0 - Y_{j-1}^0\right)&\\ + (1-\alpha_K)F_{j-1} + \alpha_K F_{j-1}^0 + (1-\alpha_K)F_j + \alpha_K F_j^0 = 0&\end{aligned} \tag{7.98}$$

and (7.97) may be rewritten as

$$Y_j = Y_j^0 + \Delta t\left((1-\gamma)\dot{Y}_j^0 + \gamma\dot{Y}_j\right). \tag{7.99}$$

Eqs. (7.98) and (7.99) become a root-finding problem of the form $F(X) = 0$, where X contains variables for each Y_j and $\dot{Y}_j$.

In the computations to come, the root finding problem of eqs. (7.98) and (7.99) was solved using Python's ScyPy optimisation module. The function used is a wrapper that uses FORTRAN-MINPACK's `hybrd` and `hybrj` algorithms [111].

7.8.5 PDE PROBLEM FORMULATION AND SOLUTION

The discretised PDE problem of (7.96) requires the definition of 8 variables per node. This means that the problem has $8(N+1)$ variables and, because each mid-segment gives 8 equations per node, $8N$ equations. This requires the definition of 8 boundary conditions. These conditions will be discussed later at each case.

All computations used the following parameters, including cable properties and discretisation:

- Drag coefficient: $C_d = 1.5$.
- Cable diameter: d = 0.025 m.
- Cable density: $\rho_c = 0.7$ kg/m.
- Cable unstretched length: $L = 100$ m.
- Tension flexibility: $c_s = 0.001$.
- Bending flexibility: $b = 1000$.
- Water density: $\rho_{\mathrm{w}} = 1025$ kg/m^3.
- Gravity: $g = 9.81$ m/s^2.
- Number of cable segments: $N = 50$.

All computations start from a computed initial condition. Roughly, the procedure is as follows:

1. Use a shooting algorithm to solve the problem of a static catenary subject to weight and drag. This model has tension as the only internal load and includes axial strain. The inputs of the problem are the position of the two hanging points and a measure of the proportion between drag and weight. The shooting algorithm is initiated from a polynomial meta-model obtained from data of previously computed cases, *i.e.*, the initial guess of initial conditions is obtained from the meta-model. This algorithm calculates the cable's geometry and distribution of tension.
2. Calculate an estimate of curvature and shear force using discrete derivatives and the previously computed geometry.
3. Solve a finite differences representation of the static problem including tension and bending, along with drag and weight. The initial guess is obtained from the calculations obtained before.

7.8.6 CASE 1

In the first case considered, the left boundary has prescribed motion and the right boundary is free to move. Also, at both ends a no-moment condition is considered. This means that at the left the following boundary conditions are used: $u(0) = u_0$, $w(0) = w_0$, $\Omega(0) = 0$, $x(0) = x_0$, and $z(0) = z_0$. At the right boundary the following conditions are used: $\Omega(L) = 0$, $S_n(L) = 0$, and $T(L) = 0$.

This formulation is used to compute two scenarios:

- The cable hangs on a fixed point; the motion is forced by the incoming current.
- To the previous scenario two control forces are added at the mid- and end-node. Also, a proportional control law is added to force the two control points to prescribed desired positions.

At both cases the left hanging point coordinates are $x_0 = 0$ and $z_0 = 0$ for any time. Also, for both scenarios the right hanging coordinates at $t = 0$ has coordinates $x_N = 0$ and $z_N = 90$ m.

For the first scenario the cable hangs on a fixed point and the motion is forced by the incoming current. Here, the results of applying a uniform-constant current of $V_c = \begin{bmatrix} -1.0 & 0.0 \end{bmatrix}^T$ m/s are shown. The resultant behaviour of the cable profile is shown in Fig. 7.34 for a total of 60 s of simulation time; each profile is drawn every 1 s.

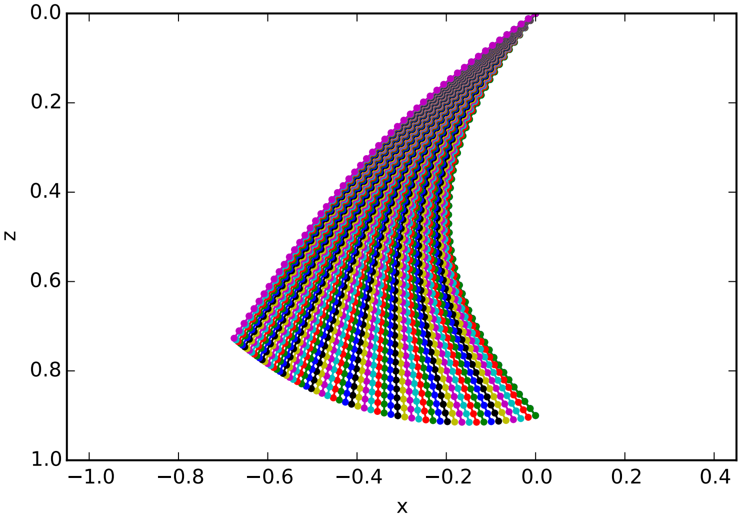


Figure 7.34. Profile variation at free drift

For the second scenario a uniform-constant current of $V_c = \begin{bmatrix} -0.5 & 0.0 \end{bmatrix}^T$ m/s is used. Here, it is desired that the mid-point of the cable reaches the coordinates

$$P_{N/2} = \begin{bmatrix} -0.10 & 0.45 \end{bmatrix}^T$$

and the end-point the coordinates

$$P_N = \left[\begin{array}{cc} 0.20 & 0.80 \end{array} \right]^T,$$

both in non-dimensional representation. To accomplish this, the proportional control law is

$$F_c = K_P(P_d - P),$$

there F_c is the applied force to each point, P_d is the desired position, P is the actual position, and K_P the proportional gain. For the two control points of this problem a high-gain value of

$$K_P = \left[\begin{array}{cc} 100 & 0 \\ 0 & 100 \end{array} \right]$$

was used. The resultant behaviour of the cable profile is shown in Fig. 7.35 for a total of 20 s of simulation time; each profile is drawn every 1 s.

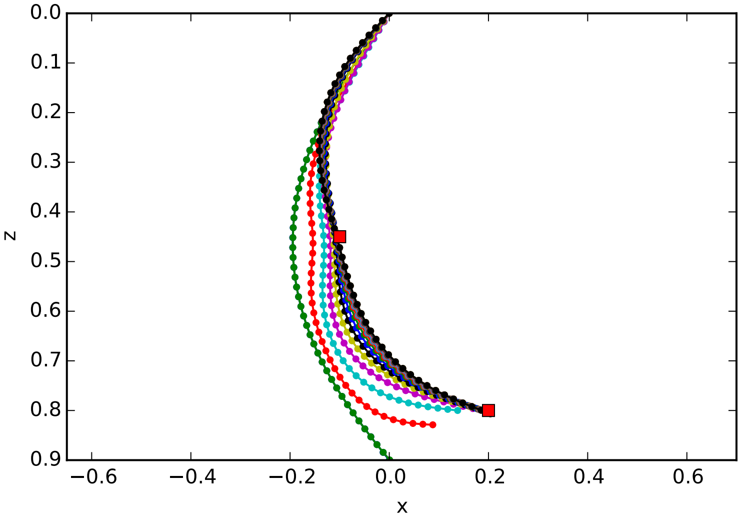


Figure 7.35. Profile variation at a two-control-point experiment

7.8.7 CASE 2

In this case both boundaries have prescribed motion. This means that at the left the following boundary conditions are used: $u(0) = u_0$, $w(0) = w_0$, $x(0) = x_0$, and $z(0) = z_0$. At the right boundary the following conditions are used: $u(L) = u_N$, $w(L) = w_N$, $x(L) = x_N$, and $z(L) = z_N$.

This formulation is used to compute one scenario: the cable hangs on two fixed points; the motion is forced by varying incoming current modelled as a 1st order Gauss-Markov Process. At this scenario the following general conditions were used:

- Left hanging point coordinates: $x_0 = 0$ y $z_0 = 0$.
- Right hanging point coordinates: $x_N = 0$ y $z_N = 90$ m.

Also, the current variation during the simulation and the cable profile are shown in Fig. 7.36

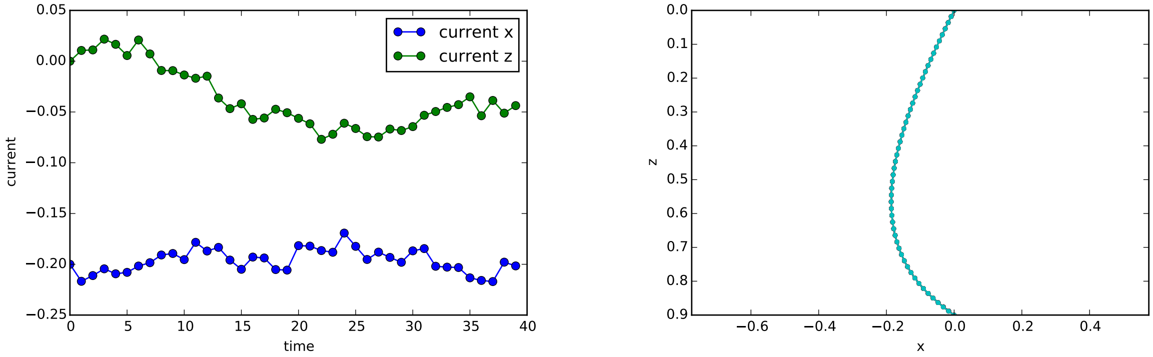


Figure 7.36. Current variation at each time step and cable profile

7.9 CONCLUDING REMARKS

This chapter approached the problem of modelling a cable that potentially connects a submerged ROV with a (stationary) point in the surface. It posed that the starting point to build-up knowledge in this matter is by considering the case of the classical catenary first and, from that point on, construct more complicated cases. Considering this, the first model was that of an inextensible catenary that bears a constant force field per unit length, such as weight or weight in water; rather conveniently, this model has analytical solution under some conditions. This simple case was extended then to the more complicated one where hydrodynamic loads such as drag and added mass are considered. The type of model that was considered last is that of a cable that includes as well elastic effects such as tension and bending. In brief, the compendium of models herein are useful as a first approach to ROV cable modelling that allow one to further compute simple motion-feasibility-related problems, as well as to serve as a preamble to studying more complex or elaborated modelling theories.

As mentioned, this work approached the modelling of an ROV cable using theories suited for highly flexible elastic rods, by building from what are considered first steps. This meant that some considerations of the model are founded on classical, simpler concepts. Geometry was defined by using Euclidean geometry, opposed to the so-called geometrically exact approaches that use differential manifolds. This can be evidenced by the description of position by vectors and orientation by Euler angles. The theories applied here privilege those that consider the cable a rod that is inextensible longitudinally, perfectly flexible under bending, and completely rigid under torsion. Lastly, numerical schemes favour the finite differences method and slightly address the finite elements method.

The catenary as scheme for modelling the cable was used extensively. The analytic solution, accompanied by a shooting algorithm, was studied first as a means to come up with an algorithm that is able to compute initial conditions; this initialisation algorithm is useful in further more complex numerical schemes on the non-linear models. In the process, the catenary was studied comprehensively, including computing some numerical results on the analytic solution, computing the problem's domain, definition of an energy function related to the solution of the numerical problem, computation of the geometry in some illustrative cases, computing regression formulas for an initial guess of the initial conditions, and, finally, approaching the problem of computing an optimal catenary configuration.

For a catenary that bears weight and drag, the three approaches to solve the model were a shooting algorithm for the ODE, finite differences, and finite elements, both in static and dynamic conditions. This finite differences method discretises the differential equation with respect to the mid-segment using first-order centred differences. This formulation leads to a tangent linear system, for where the Jacobians of the drag function were the more complex terms. For the dynamics case, the tangent linear system is compatible with a Newmark-β time integration scheme. Conversely, the finite elements method used Galerkin's convention of weighted residuals, as well as a mixed formulation that assigns the same order of approximation to internal force and position.

The chapter closed by performing computations on the WHOI cable model, where the equations are specified using cable frame coordinates. These models were computed using the finite differences method, where the differential equation is discretised with respect to the mid-segment using first-order centred differences. In the computations, the variables were written in non-dimensional form. The computations on the statics model were performed to study the effect of different elasticity variables, such as those related to tension and bending. Conversely, the computations on the dynamics model were carried out to study free drift motion and propose a prototype concept for a cable that has two position-controllable points.

Altogether, this chapter laid a collection of methods from the perspective of solving the question, how to start modelling and computing the effect of a cable in an ROV? This approach gives way to myriad routes for improvement: some of these routes are discussed next. The way numerical methods are computed should be dealt with in detail, such as domain discretisation, equation discretisation schemes, convergence conditions, and so on. This work used Euclidean geometry, a route for further work may be through using differential manifold geometry. Elasticity was confronted, but it needs to be tackled even further. Experimental validation of the numerical models needs to be addressed. Finally, methods for connecting the computation of the cable and the ROV's time-domain dynamics requires dedicated work, because the models use different time and space scales, *e.g.*, the cable often requires smaller time resolution.

Part III

Manoeuvring and motion feasibility

Chapter 8

Time-domain simulation framework

The goal of time-domain simulation is to solve model (2.5) and related equations. This means to predict future behaviour from known initial conditions. In the context of this work, this fundamentally means to predict the trajectory of state $\mathbf{q}(t)$, *i.e.*, the time vehicle's configuration (position and orientation) η and velocity ν. Time domain-simulation requires considering two main components: first, conceptualising simulations under the idea of object-oriented programming, and, second, defining discrete-time integration algorithms. The former is mostly useful to organise and configure simulation scenarios and the latter to numerically compute the trajectory of state, $\mathbf{q}(t)$, for such scenarios.

8.1 OBJECT-ORIENTED SIMULATION FRAMEWORK

Object-oriented programming is a programming paradigm where objects are instances (a type of variable) that have the ability to store data and contain routines, namely variables and methods. The data contain parameters and internal processing variables, and the methods contain algorithms that perform useful computations. In this context, objects represent modelling blocks, its variables model parameters and useful variables, and its methods compute models and other useful subroutines.

A starting point for developing a simulation framework using object-oriented idiosyncrasies is model (2.5). Model (2.5) is shown again in Fig. 8.1 and their components indicated, namely rigid body, hydrodynamics, hydrostatics, actuators, cable, and others. The main concept is to treat each of those components as objects. This means to define each model block as an object; then, each object is able to store model parameters and compute model-related subroutines. For instance, the rigid body block requires one to declare the vehicle's mass, moments of inertia, and centre of gravity; furthermore, its methods should include routines, *e.g.*, such as those to compute mass matrix $\mathbf{M}_{RB}$ and Coriolis matrix $\mathbf{C}_{RB}$. Consequently, a simulation framework for vehicle motion should conceptualise rigid body, hydrodynamics, hydrostatics, actuators, and cable components as individual objects and their interactions.

$$\underbrace{\mathbf{M}_{RB}\dot{\nu} + \mathbf{C}_{RB}\left(\nu\right)\nu}_{\text{RIGID BODY}} + \underbrace{\mathbf{M}_{A}\dot{\nu}_r + \mathbf{C}_{A}\left(\nu_r\right)\nu_r + \mathbf{D}\left(\nu_r\right)\nu_r}_{\text{HYDRODYNAMICS}} + \underbrace{\mathbf{g}\left(\eta\right)}_{\text{HYDROSTATICS}} = \underbrace{\tau(\mathbf{u})}_{\text{ACTUATORS}} + \underbrace{\tau_{\text{cable}}}_{\text{CABLE}} + \underbrace{\tau_{\text{wave}}}_{\text{WAVE}}.$$

Figure 8.1. ROV model's blocks

A more complete development of the idea of a motion simulation framework should include as well other motion-related components such as thrust allocation, control, and environmental behaviour. This idea is illustrated in Fig. 8.2. Here, an ambiance object, namely *scenario*, is defined. The scenario conceptually defines everything that is needed to numerically solve model (2.5) and related equations. Consequently, the scenario object includes simulation parametrisation such as time-integration algorithm and subsequent configuration parameters, as well as the system's component blocks. In other words, the scenario is a wrapper of everything required to simulate, and, as shown in Fig. 8.2, these fundamental components are the ROV system and environment.

In Fig. 8.2, the ROV system object is a wrapper of vehicle, cable, thrust allocation, and control objects. Conversely, the environment object fundamentally includes current and density objects. These last two provide means to compute current ν_c and density ρ_{W} to most objects' computations. Among the ROV system's objects, the vehicle object is a wrapper of rigid body, hydrodynamics, hydrostatics, and actuators. All these objects are included in model (2.5) and previously described in Fig. 8.1. The rigid body object is of particular interest because conceptually the ROV motion problem is the classic physics problem of integrating rigid motion dynamic equations. In the end, all objects different to rigid body object provide information to the rigid body object.

In Fig. 8.2, thrust allocation, control, and cable objects are treated separately from the ROV system object. Thrust allocation and control objects are treated separately because, conceptually, they represent algorithms to be implemented into the ROV's computer. The cable object is treated separately because it requires solving a different dynamics problem,

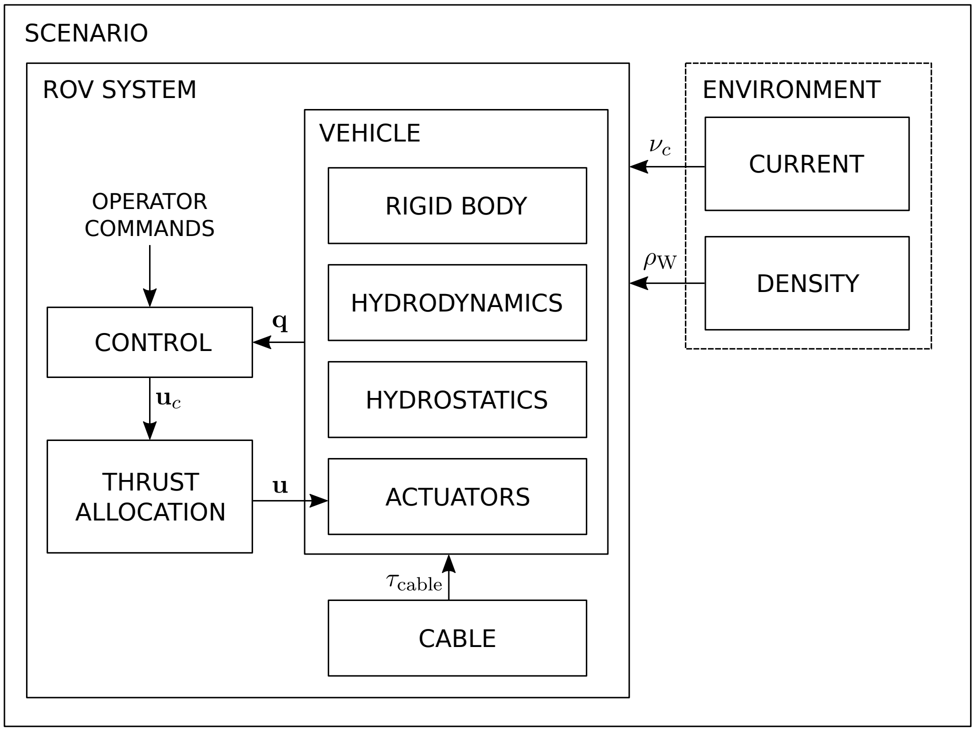


Figure 8.2. Time-domain simulation object structure

i.e., solving any type of elastic rod model. Thrust allocation and control are the elements that convert some sort of operator commands into thrusters commands. First, the control object converts operator commands into control commands $\mathbf{u}_c$. Second, the thrust allocation object converts the control commands into thrusters commands $\mathbf{u}$. Finally, the control commands alter the way actuators operate. As mentioned, the cable object requires solving different dynamics in often different time and space scale resolutions. Because of this, the cable model is computed separately to the rigid body dynamics problem. After the cable model is computed, the coupling with the rigid body model is done through forces and moments vector τ_{cable}.

The framework in Fig. 8.2 shows everything that needs to be specified in order to be able to simulate, in the sense that each of these objects and their parameters need to be declared. Additionally, it shows what kind of data needs to be exchanged among objects in order to pursue computations. Nevertheless, it is necessary to further specify how computations are performed and how data is intechanged.

8.2 TIME INTEGRATION

The main objective of time-domain simulation is to construct time trajectories of the state, *i.e.*, $\mathbf{q}(t)$. As mentioned, the state is defined as a Cartesian product of vehicle's configuration (position and orientation) and velocity, *i.e.*, $\mathbf{q} = \begin{bmatrix} \eta^{\mathsf{T}} & \nu^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$. The space defined by $\mathbf{q}$ is called the state space. At each $\mathbf{q}$ a tangent space of vectors $\dot{\mathbf{q}} = \begin{bmatrix} \dot{\eta}^{\mathsf{T}} & \dot{\nu}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ could be defined. These spaces are important because kinematic and dynamic models (2.2) and (2.5), namely

$$\mathbf{M}_{RB}\dot{\nu} + \mathbf{C}_{RB}\left(\nu\right)\nu + \mathbf{M}_{A}\dot{\nu}_r + \mathbf{C}_{A}\left(\nu_r\right)\nu_r + \mathbf{D}\left(\nu_r\right)\nu_r + \mathbf{g}\left(\eta\right) = \tau(\mathbf{u}) + \tau_{\text{cable}} + \tau_{\text{wave}}.$$

and

$$\dot{\eta} = \mathbf{J}(\eta)\nu,$$

are defined as restrictions at those spaces. In other words, models (2.2) and (2.5) impose the restrictions needed to uniquely find a trajectory $\mathbf{q}(t)$. For instance, assuming that $\dot{\nu} = \dot{\nu}_c$ and using a different order, one can write model (2.5) as

$$\left(\mathbf{M}_{RB} + \mathbf{M}_{A}\right)\dot{\nu} = -\left(\mathbf{C}_{RB}\left(\nu\right)\nu + \mathbf{C}_{A}\left(\nu_r\right)\nu_r + \mathbf{D}\left(\nu_r\right)\nu_r + \mathbf{g}\left(\eta\right)\right) + \tau(\mathbf{u}) + \tau_{\text{cable}}.$$

If one defines the total mass matrix $\mathbf{M}_T = \mathbf{M}_{RB} + \mathbf{M}_A$ and a total load vector

$$\mathbf{f}_T(\eta,\nu) = -\left(\mathbf{C}_{RB}\left(\nu\right)\nu + \mathbf{C}_{A}\left(\nu_r\right)\nu_r + \mathbf{D}\left(\nu_r\right)\nu_r + \mathbf{g}\left(\eta\right)\right) + \tau(\mathbf{u}) + \tau_{\text{cable}},$$

the model becomes the rigid body motion equation

$$\mathbf{M}_T\dot{\nu} = \mathbf{f}_T(\eta,\nu).$$

Finally, assuming $\mathbf{M}_T$ is invertible, one obtains.

$$\dot{\nu} = \mathbf{M}_T^{-1}\mathbf{f}_T(\eta,\nu). \tag{8.1}$$

This rigid body motion equation and kinematic transformation (2.2), $\dot{\eta} = \mathbf{J}(\eta)\nu$, explicitly pose a relation between the state and vectors in its tangent space.

Because the resultant equations are in general not solvable using analytical methods, the computation of trajectories is done discretely. This means that time becomes the sequence $t = \{t_0, t_1, \ldots, t_k, t_{k+1}, \ldots\}$, with the time-step given by $\Delta t = t_{k+1} - t_k$ for $k = 0, 1, \ldots$. This implies that variables are evaluated at discrete times and defined as $\eta_k = \eta(t_k)$, $\nu_k = \nu(t_k)$, and $\mathbf{q}_k = \mathbf{q}(t_k)$. Hence, the discrete-time integration problem becomes finding state $\mathbf{q}_{k+1}$, at $t = t_{k+1}$, from state $\mathbf{q}_k$, at $t = t_k$. To accomplish this, two main types of algorithms (or schemes) could defined: implicit and explicit.

8.2.1 EXPLICIT SCHEMES

Integration schemes are often defined in vector spaces using the idea that the next-time variable, *e.g.* $\mathbf{q}_{k+1}$, is obtained as the previous time variable, *e.g.* $\mathbf{q}_k$, plus some increment, *e.g.* $\Delta\mathbf{q}$; this is $\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta\mathbf{q}$. In explicit schemes, the increments are dependent only on previous-time variables, *i.e.*, $\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta\mathbf{q}(t_k)$. As the name suggests, the next-time variable is explicitly computed from previous-time variables exclusively. The most immediate scheme is a first-order Euler algorithm where derivatives are replaced by a first-order forward difference, *i.e.*, $\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta t\, \dot{\mathbf{q}}(t_k)$. More explicitly, for state components, namely configuration and velocity, this is

$$\begin{aligned}\eta_{k+1} &= \eta_k + \Delta t\, \dot{\eta}(t_k),\\ \nu_{k+1} &= \nu_k + \Delta t\, \dot{\nu}(t_k).\end{aligned}$$

By using kinematic transformation (2.2) and rigid-body equation (8.1), the integration scheme becomes

$$\begin{aligned}\eta_{k+1} &= \eta_k + \Delta t \mathbf{J}_\Theta(\eta_k)\nu_k,\\ \nu_{k+1} &= \nu_k + \Delta t \mathbf{M}_T^{-1}\mathbf{f}_T(\eta_k, \nu_k).\end{aligned}$$

8.2.2 IMPLICIT SCHEMES

In implicit schemes the increment is dependent also on next-time variables, *i.e.*, $\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta\mathbf{q}(t_k, t_{k+1})$. In this case, next-time variables appear on the right side of the equation as well. This means that the next-time variable appears implicitly in the equation. This now requires the solution of an equation system, rather than a straight-forward vector addition. The most straight-forward explicit scheme is a first-order Euler algorithm where derivatives are replaced by a first-order backward difference, *i.e.*, $\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta t\, \dot{\mathbf{q}}(t_{k+1})$. In this case, velocity integration becomes

$$\nu_{k+1} = \nu_k + \Delta t\, \dot{\nu}(t_{k+1}) = \nu_k + \Delta t\, \dot{\nu}_{k+1}. \tag{8.2}$$

To complete the algorithm, in model (2.5), η, ν and $\dot{\nu}$ are evaluated at time t_{k+1}, *i.e.*,

$$\mathbf{M}_T\dot{\nu}_{k+1} = -\left(\mathbf{C}_{RB}\nu_{k+1} + \left(\mathbf{C}_A + \mathbf{D}\right)\left(\nu_{k+1} - \nu_c\right) + \mathbf{g}\left(\eta_{k+1}\right)\right) + \tau + \tau_{\text{cable}}.$$

Please note that the definition of relative velocity, $\nu_r = \nu - \nu_c$, was used. Combining this and (8.2) the velocity update algorithm becomes the equation

$$\left[\mathbf{M}_T + \Delta t(\mathbf{C}_{RB} + \mathbf{C}_A + \mathbf{D})\right]\nu_{k+1} = \mathbf{M}_T\nu_k + \Delta t\left[\left(\mathbf{C}_A + \mathbf{D}\right)\nu_c - \mathbf{g} + \tau + \tau_{\text{cable}}\right].$$

This equation defines next-time velocity ν_{k+1} implicitly. A common way to solve the problem is to compute η_{k+1} explicitly and then ν_{k+1} implicitly.

8.2.3 SOLVERS IN PYTHON

More elaborated implicit and explicit schemes are already implemented in numerical computing software. For instance, Python has the ScyPy module that contains an implementation of ordinary differential equation (ODE) solvers, with options such as [110]

- `vode`, a real-valued variable-coefficient ODE solver with fixed-leading-coefficient implementation, which includes implicit Adams method and an explicit method based on backward differentiation formulas;
- `lsoda`, a real-valued variable-coefficient ODE solver with fixed-leading-coefficient implementation, with an automatic method for switching between implicit Adams method and backward differentiation formulas; and,
- `dopri5`, an explicit Runge-Kutta method of order 4-5 that includes step-size control and dense output.

Chapter 9

Manoeuvring time-domain simulations

This chapter focuses on performing time-domain simulations using an ROV model to evaluate its manoeuvring performance; consequently, it could be understood as a sequel of Chapter 6, namely ROV hydrodynamics, where the concepts of Chapter 8, time-domain simulation, are applied. By following this idea, manoeuvring time-domain simulations are regarded here as a first approach to study ROV motion feasibility. This is done through the verification of the performance of an ROV at different scenarios. The main advantage of pursuing this approach is that motion performance can be evaluated qualitatively and quantitatively at various conditions. The simulations become then a tool to understand the ROV's manoeuvring and make further design-, control-, and operation-related decisions.

The approach applied in this chapter uses VISOR3's model to perform manoeuvring time-domain simulations and subsequent analyses. More concretely, the different simulation scenarios include the following:

- Open-loop response to actuating thrusters individually, *i.e.*, without using a thrust allocation algorithm.
- Open-loop response to actuating thrusters through a thrust allocation algorithm.
- Closed-loop response to simple (multi-loop) PID algorithms.

9.1 OPEN-LOOP, NO THRUST ALLOCATION

The first step in studying VISOR3's manoeuvrability consists in simulating the ROV's response when the thrusters are actuated at will, without any means of control. Consequently, in this section, all simulation scenarios consist in acting the thrusters selectively and individually at a constant propeller speed in order to produce simple manoeuvres. The simulations herein are grouped as calm-water and oblique-current manoeuvres.

9.1.1 CALM-WATER MANOEUVRES

Calm-water manoeuvres, where current is (exactly) zero, are regarded here as the first to approach because they are the simplest. The different manoeuvres are intended first to produce motion along the four controllable degrees of freedom individually and, second, to produce circular manoeuvres on the xy-plane.

9.1.1.1 Surge-direction thrust

In order to perform surge-direction motion, the same input is assigned to port and starboard thrusters. In this case, different propeller speeds are tested to check forward and backward motion, namely for propeller speeds of −500, −250, 250, and 250 rpm. The simulations are run for 30 s and the results are illustrated in Figs. 9.1 and 9.2.

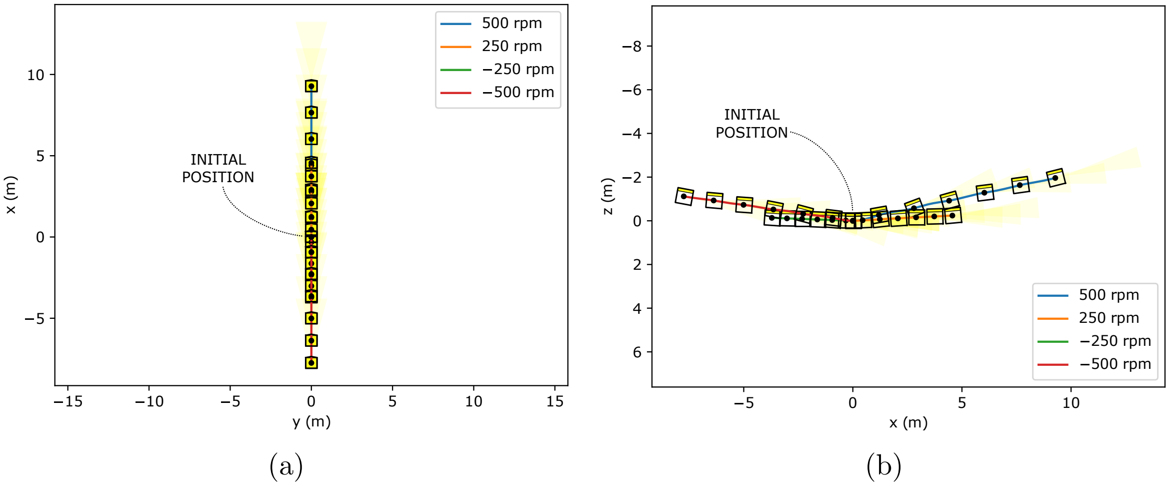


Figure 9.1. Longitudinal thrust at calm water and different propeller speeds: trajectories

Figure 9.1 shows the ROV trajectory at the different propeller speeds. Figure 9.1 (a) shows the projection onto the xy-plane, *i.e.*, a top-view projection of the motion; conversely, Fig. 9.1 (b) shows the projection onto the xz-plane, *i.e.*, a side-view projection of the motion. From Fig. 9.1 (a) it becomes apparent that the ROV is able to keep its heading constant when not perturbed by current. Nevertheless, from Fig. 9.1 (b), it becomes apparent that there is a lift-force effect that alters vertical motion and increases with speed. Additionally, it can be noticed that there is more advance in forward motion that in backward motion: this reflects the propeller's nature, not the hydrodynamic meta-model's nature (as the model assumed symmetry).

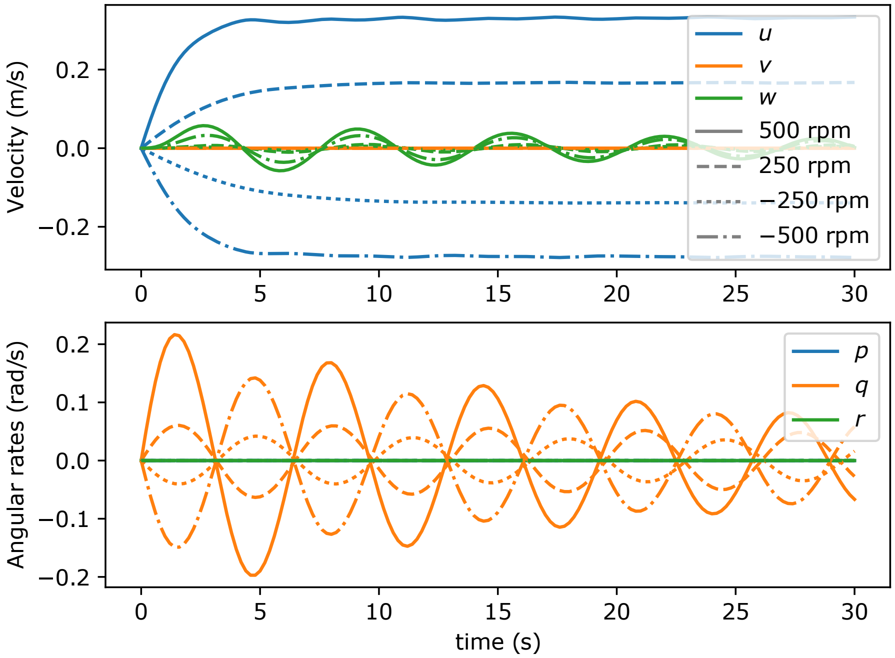


Figure 9.2. Longitudinal thrust at calm water and different propeller speeds: kinematic variables

Figure 9.2 further shows the behaviour of kinematic variables, *i.e.*, linear velocity components u, v, and w and angular velocity components (angular rates) p, q, and r. From the figure the coupling among surge, heave, and pitch becomes evident: surge motion alters pitch, and changes in pitch produce a lifting force that alters vertical motion (heave). This condition suggests that proper surge motion control requires addressing pitch and heave.

9.1.1.2 Sway-direction thrust

To perform sway-direction motion, a constant input is assigned to the lateral thruster. As previously, different propeller speeds are tested to check both directions of motion, for propeller speeds of −500, −250, 250, and 250 rpm. The simulations are run for 30 s and the results are illustrated in Figs. 9.3 and 9.4.

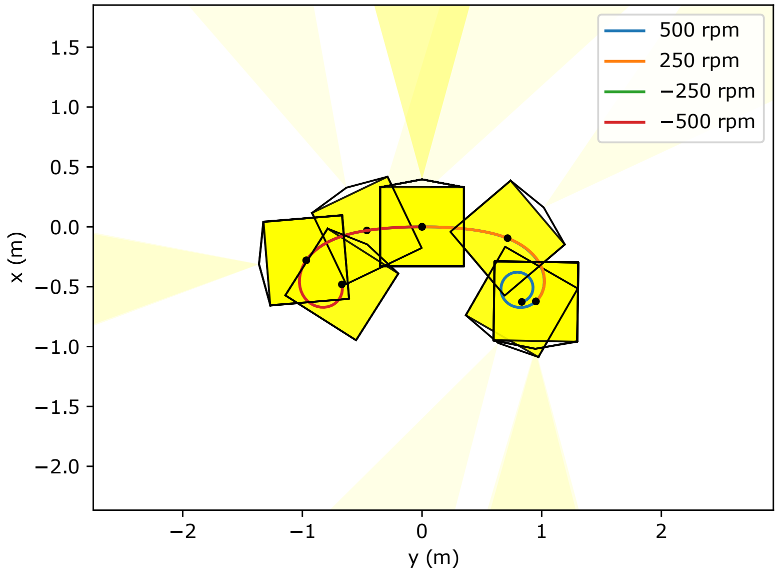


Figure 9.3. Lateral thrust at calm water and different propeller speeds: trajectories

Figure 9.3 shows an xy-plane projection of the trajectory for the different propeller speeds. From the figure, it becomes apparent that the ROV does not keep straight lateral motion to the point of producing a small-radius circular trajectory. This behaviour can be attributed mainly to the fact that the transversal thruster is displaced towards the bow. Furthermore, Fig. 9.4 shows the behaviour of linear and angular velocity components. From the plots in this figure, it becomes apparent that all degrees of freedom get excited; nevertheless, controlling yaw is the most relevant condition to be considered.

9.1.1.3 Heave-direction thrust

To perform heave-direction motion, a constant input is assigned to the vertical thruster. As usual, different propeller speeds are tested to check both directions of motion, for propeller speeds of −500, −250, 250, and 250 rpm. The simulations are run for 30 s and the results are illustrated in Figs. 9.5 and 9.6.

Figure 9.5 shows an xz-plane projection of the trajectory for the different propeller speeds. In the figure the trajectory shown is approximately straight in all cases. More specifically, Fig. 9.6 shows how vertical motion excites roll motion significantly and pitch motion in a smaller degree; moreover, as a consequence of roll excitation, sway gets excited as well.

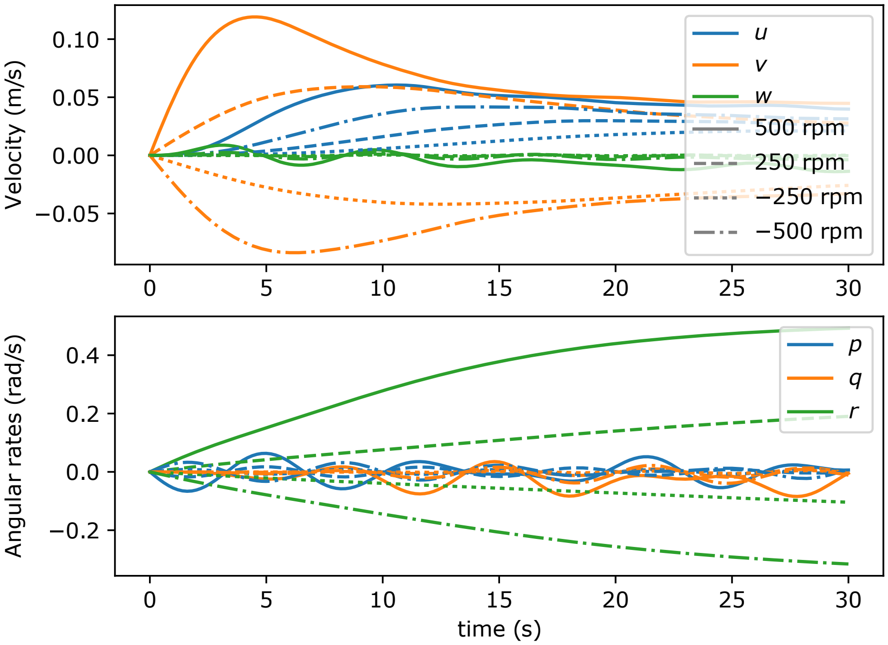


Figure 9.4. Lateral thrust at calm water and different propeller speeds: kinematic variables

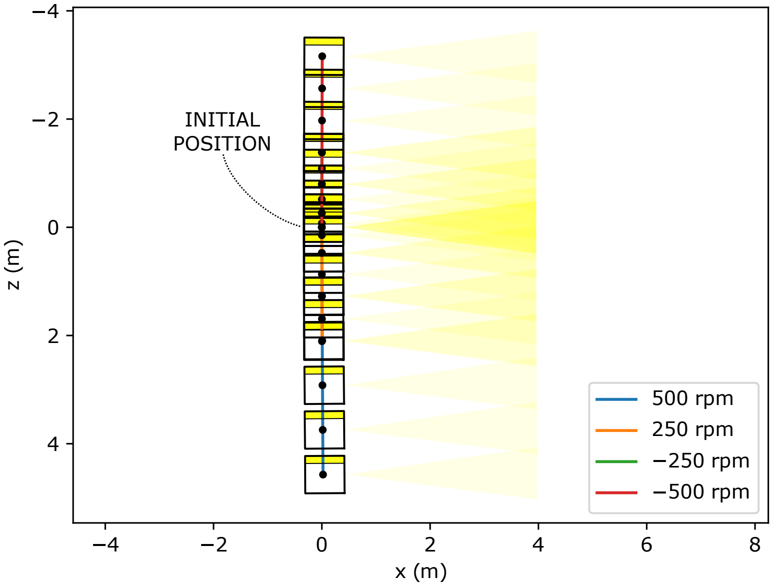


Figure 9.5. Vertical thrust at calm water and different propeller speeds: trajectories

9.1.1.4 Yaw-inducing thrust

To induce yaw rotation, constant inputs are assigned to port and starboard thrusters with opposing-direction thrust. As in the previous cases, different propeller speeds are tested to

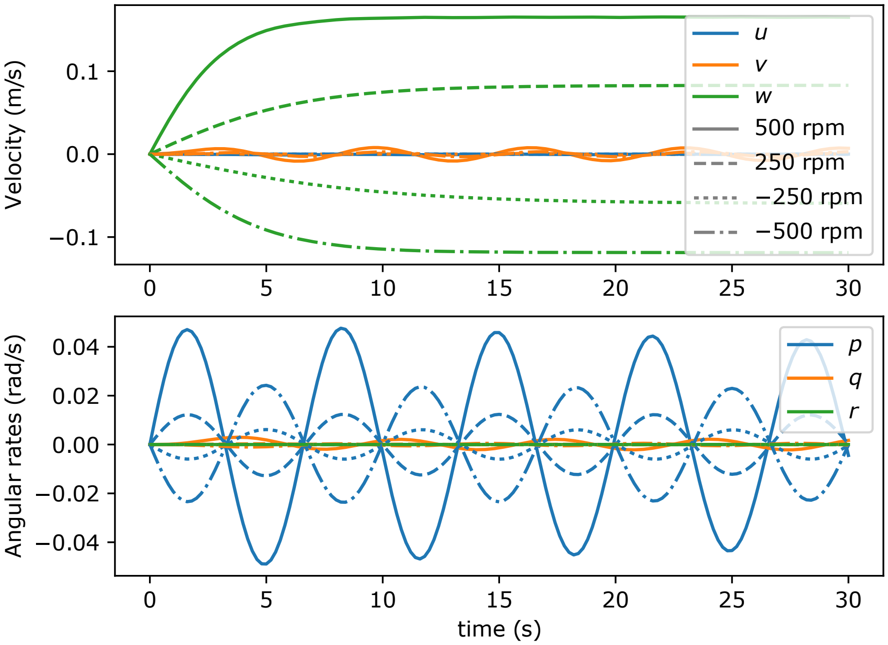


Figure 9.6. Vertical thrust at calm water and different propeller speeds: kinematic variables

check both directions of rotation, for propeller speeds of 250 and 250 rpm. The simulations are run for 30 s and the results are illustrated in Fig. 9.7.

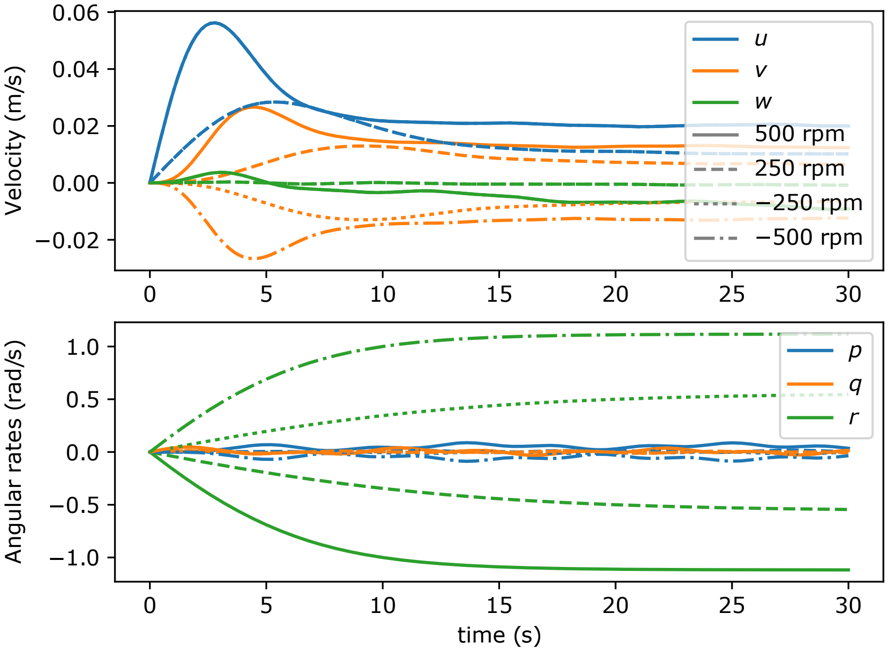


Figure 9.7. Yaw-inducing thrust at calm water and different propeller speeds: kinematic variables

From the kinematic variables shown in Fig. 9.7 it becomes apparent that yaw motion excites all degrees of freedom in a small amount. This is attributed mainly to the propellers' disparity between forward and backward thrust, which produces a non-zero linear velocity; this condition is susceptible of being attenuated by a thrust allocation algorithm.

9.1.1.5 Circular-trajectory-inducing thrust

To perform circular trajectories, constant inputs are assigned to port and starboard thrusters with differential thrust. To do this, starboard thruster speed is 250 rpm and port thruster speed is a fraction of starboard thruster's, namely in percentages of 90, 93, 94, and 95 %. The simulations are run for 120 s and the results are illustrated in Figs. 9.8 and 9.9.

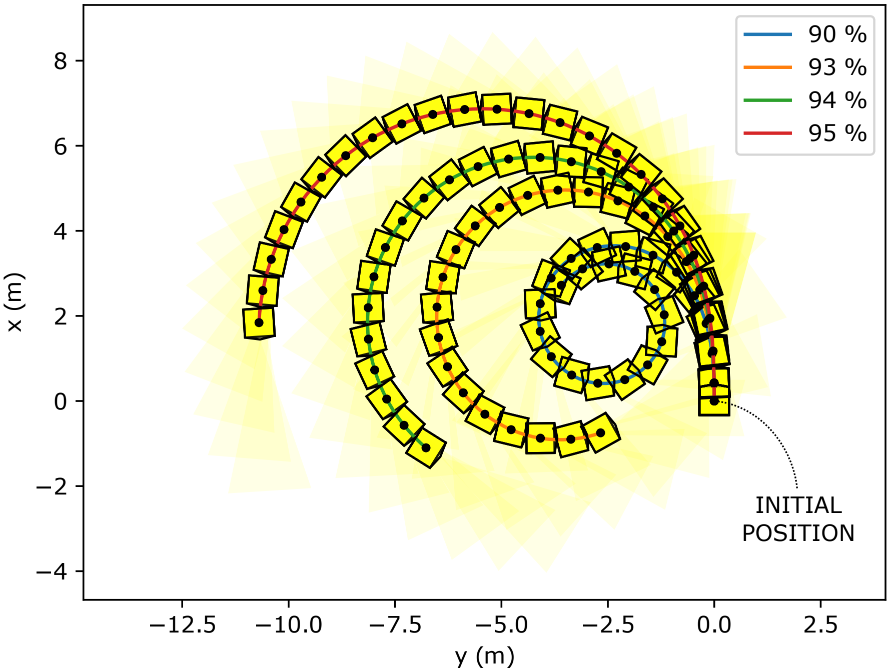


Figure 9.8. Circle-inducing thrust at calm water and different propeller speeds: trajectories

Figure 9.8 shows an xy-plane projection of the trajectory for the different simulations. In the figure it is apparent how (even) this small disparity in thrust produces the different circular trajectories. Furthermore, in Fig. 9.9 different phenomena are present: yaw rate stabilises in a constant value related to the turning radius; sway velocity is non-zero, indicating horizontal drift; surge motion, as usual, induces pitch and, consequently, a lift force that produces heave; and, finally, roll gets excited to sustained oscillations.

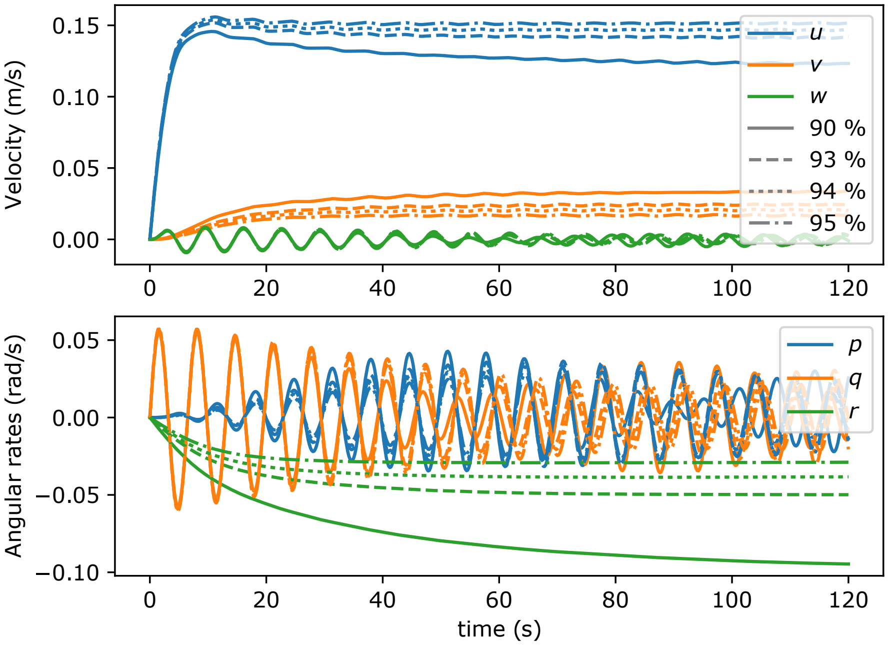


Figure 9.9. Circle-inducing thrust at calm water and different propeller speeds: kinematic variables

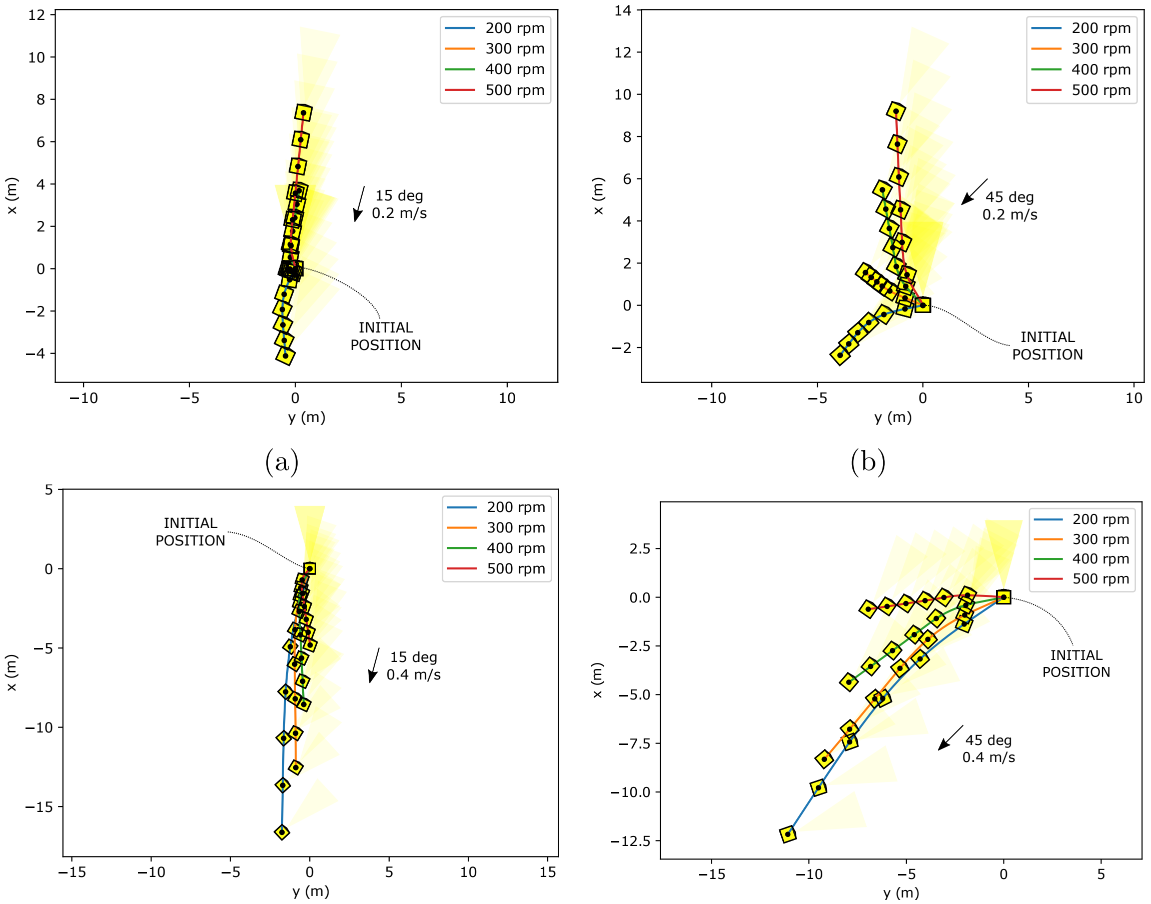
9.1.2 OBLIQUE-CURRENT MANOEUVRES

The oblique-current manoeuvres considered here concentrate on performing surge-direction thrust at non-zero current. Different thrust conditions are tested by assigning different constant propeller speeds to port and starboard thrusters; conversely, different current conditions are tested by assigning different incoming current magnitudes and directions. The scenarios are assembled by testing all combinations of the following input parameters:

- Propellers speeds of 200, 300, 400, and 500 rpm.
- Current magnitudes of 0.2 and 0.4 m/s, representing near-calm and light-current conditions.
- Current directions of 15 and 45 deg.

The results from all these scenarios are grouped according to all combinations of current conditions and illustrated by Figs. 9.10 and 9.11.

Figure 9.10 shows an xy-plane projection of the ROV's trajectory in all cases. Figures 9.10 (a) and (b) show behaviour at 0.2 m/s current. From the plots, it becomes apparent that the ROV is able to withstand the imposed current for propeller speeds greater than 300 rpm for both 15 and 45 deg current directions. Additionally, it is apparent how the ROV adopts



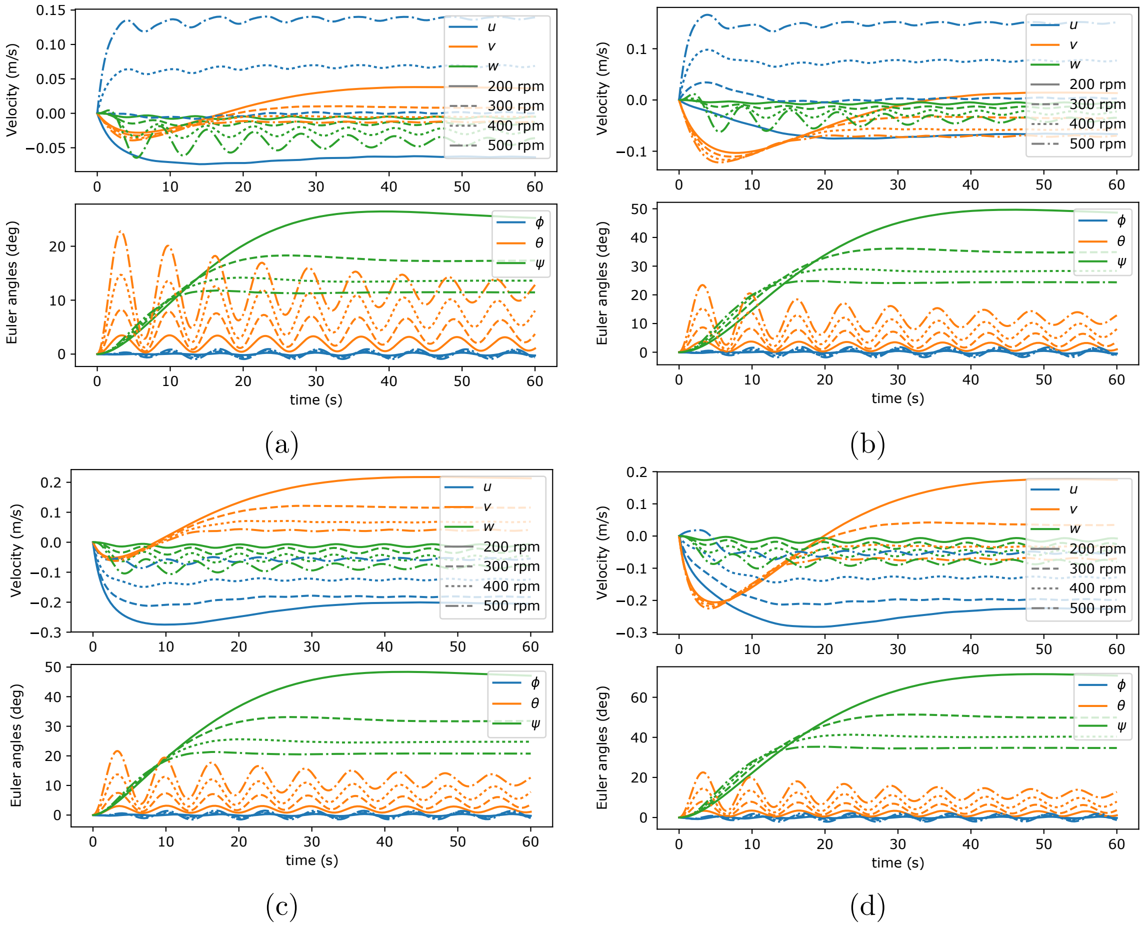
(a) (b)

(c) (d)

Figure 9.10. Surge motion in oblique-current: trajectories

certain heading that is related to the incoming current direction and, subsequently, reaches equilibrium around yaw. Simulated propeller speeds go up to 500 rpm, which are considerably below their maximum. A reason for this is that, checking the results in Figs. 9.11 (a) and (b), increasing propeller speeds produce variations of pitch angle that produce subsequent lift force. Besides the tendency of increasing pitch angle, increasing lift makes the ROV not able to keep horizontal motion.

Figures 9.10 (c) and (d) show the trajectories at 0.4 m/s current. In this case, it becomes apparent that the ROV is not able to withstand the imposed current for the simulated propeller speeds, at both 15 and 45 deg current directions: the ROV drifts backwards. As in the previous two cases, the ROV reaches certain heading that is related to the incoming current direction where equilibrium around yaw is achieved. It is possible for this ROV to simulate propeller speeds beyond 500 rpm, but, as in the previous two cases and checking the results



(a)

(b)

(c)

(d)

Figure 9.11. Surge motion in oblique-current: kinematic variables

in Figs. 9.11 (c) and (d), increasing propeller speeds produces variations of pitch angle and lift force that induce vertical motion. These results evidence the difficulties of manoeuvring VISOR3 in currents that are beyond near-calm; current magnitudes that are around 0.4 m/s are considered light and do not allow the ROV to be easily manoeuvred without further (feedback) control actions.

9.2 OPEN-LOOP, WITH THRUST ALLOCATION

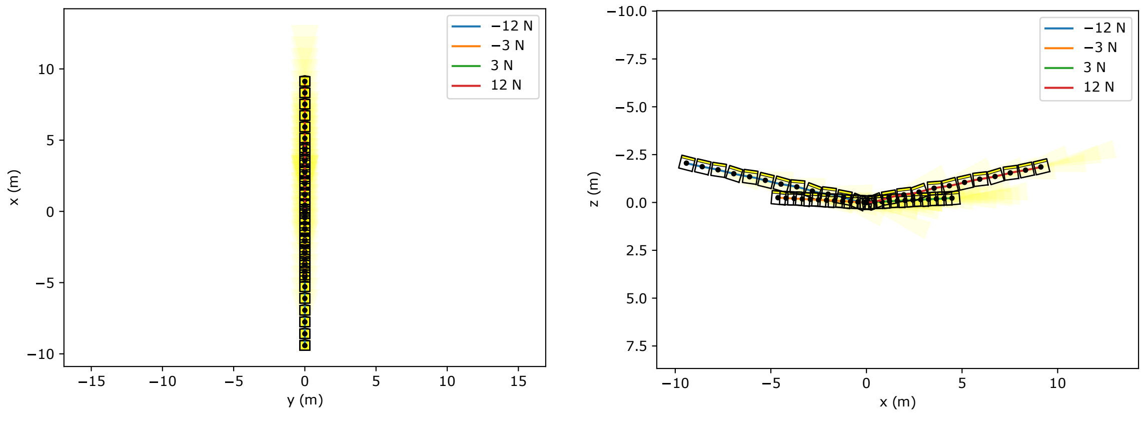
The second step in studying VISOR3's manoeuvrability consists in simulating the ROV's response when the thrusters are actuated by using a thrust allocation algorithm. Consequently, in this section, all simulation scenarios consist in acting the thrusters through the thrust allocation algorithm indicated in Sec. 3.1 at a constant input command (virtual force command) in order to produce simple manoeuvres. The simulations herein are grouped as calm-water and oblique-current manoeuvres.

9.2.1 CALM WATER, DIFFERENT PROPELLER SPEEDS

In this section, calm-water manoeuvres, where current is (exactly) zero, are intended to produce motion along the four controllable degrees of freedom, namely surge, sway, heave, and yaw, through the proposed thrust allocation algorithm.

9.2.1.1 Surge-direction command

In order to perform surge-direction motion, a virtual force command is assigned to surge input, namely for commands equivalent to forces of -12, -3, 3, and 12 N to allow forward and backward motion. The simulations are run for 30 s and the results are illustrated in Figs. 9.12 and 9.13. Figures 9.12 (a) and (b) show the ROV trajectory at the different surge commands: Fig. 9.12 (a) shows the projection onto the xy-plane and Fig. 9.12 (b) the corresponding projection onto the xz-plane. Figure 9.13 further shows the behaviour of linear velocity components u, v, and w and angular velocity components p, q, and r.



(a) (b)

Figure 9.12. Longitudinal thrust at calm water and different surge commands: trajectories

The main difference between manoeuvring with and without thrust allocation, besides easing operator's motion commands, is that the latter allows one to equate forward and backward actions by acknowledging that forward and backward thrust coefficients are different. This can be checked in Figs. 9.12 (a) and (b) by noticing that the ROV advances the same distance forward and backwards. More precisely, in this simulation a forward surge command of 3 N assigns a propeller speed of 246 rpm to port and starboard thrusters; conversely, a backward surge command of −3 N assigns a propeller speed of −305 rpm. This is more pronounced at a greater force command: a forward surge command of 12 N assigns a propeller speed of 491 rpm and the corresponding backward surge command a propeller speed of −611 rpm.

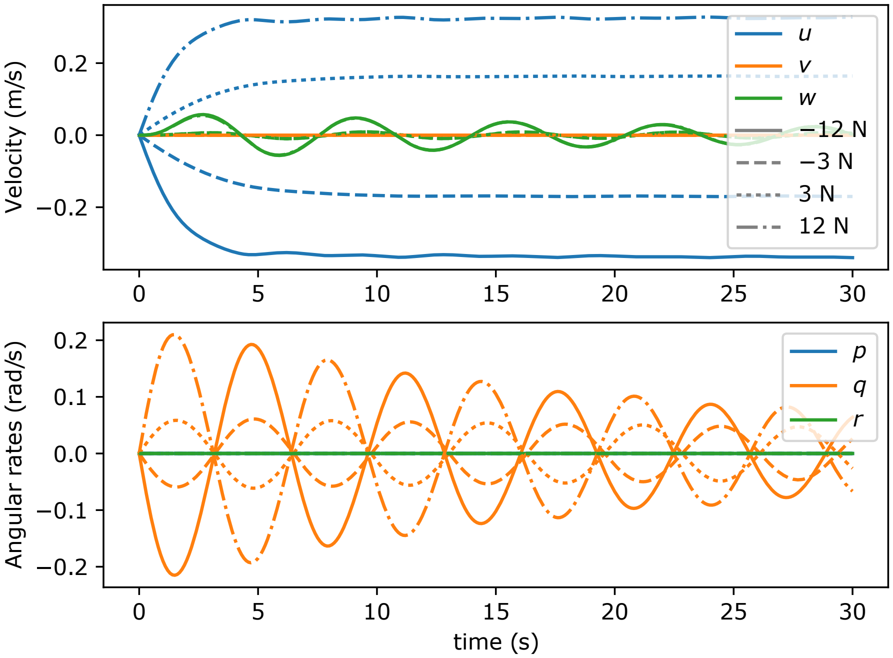


Figure 9.13. Longitudinal thrust at calm water and different surge commands: kinematic variables

From Figs. 9.12 (a) and (b) and Fig. 9.13, it becomes apparent that the features of the ROV's motion are similar to those of the non-thrust-allocation case: the ROV is able to keep its heading constant when not perturbed by current and there is a lift-force effect that alters vertical motion and increases with speed. This condition suggests that proper surge motion requires not only thrust allocation but addressing pitch and heave through feedback control.

9.2.1.2 Sway-direction command

In order to perform sway-direction motion, a virtual force command is assigned to sway input, namely for commands equivalent to forces of −12, −3, 3, and 12 N to allow port- and starboard-side motion. The simulations are run for 30 s and the results are illustrated in Figs. 9.14 and 9.15. Figure 9.14 shows a projection onto the xy-plane of the ROV trajectory at the different force commands. As usual, Fig. 9.15 further shows the behaviour of linear velocity components u, v, and w and angular velocity components p, q, and r.

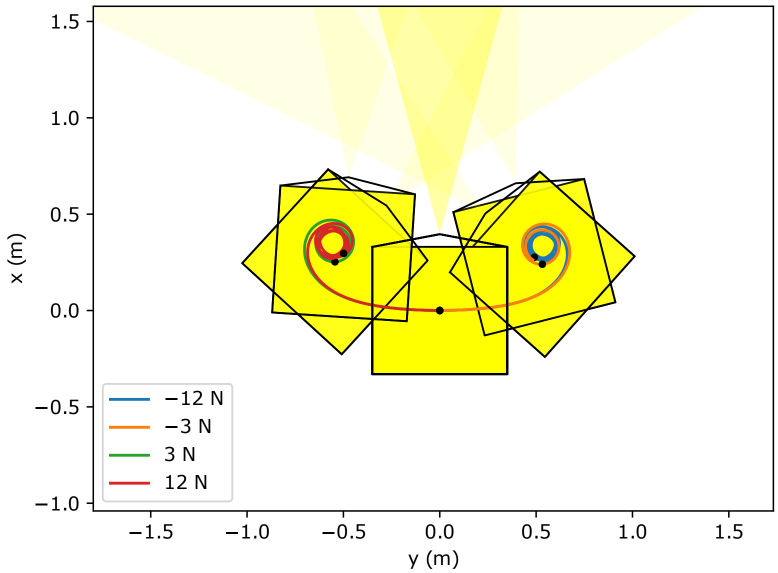


Figure 9.14. Lateral thrust at calm water and different sway commands: trajectories

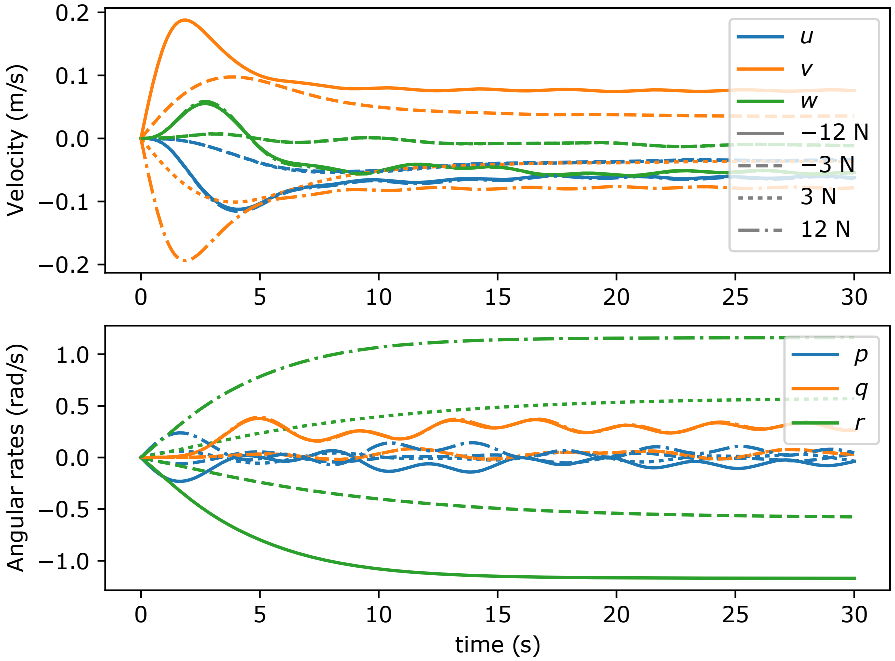


Figure 9.15. Lateral thrust at calm water and different sway commands: kinematic variables

In sway motion, the thrust allocation algorithm effect is two-fold: besides equating port- and starboard-side actions as it happened with surge motion, it performs a compensation of the yaw moment induced by the fact that the lateral thruster is displaced towards the bow. The compensation of the yaw moment induced by the lateral thruster is done by using port and starboard thrusters at opposed directions. For instance, when a sway force command of 12 N is given to the thrust allocation algorithm, it assigns 949 rpm to the lateral thruster; then, to compensate yaw moment it assigns as well 632 rpm to the port thruster and −786 rpm to the starboard thruster.

From Figs. 9.14 and 9.15 one can acknowledge the fact that a sway force command does not produce straight lateral motion, despite the aforementioned compensation of yaw performed by the thrust allocation algorithm. As in the no-thrust-allocation case, a small-radius circular trajectory is produced and, in this case, with yaw motion in the opposite direction. This effect is attributed to the behaviour of hydrodynamic yawing moment; this means that despite having a thrust allocation algorithm, feedback-controlling yaw is required to perform pure sway motion.

9.2.1.3 Heave-direction command

In order to perform heave-direction motion, a virtual force command is assigned to heave input, namely for commands equivalent to forces of −12, −3, 3, and 12 N to allow both directions of motion. The simulations are run for 30 s and the results are illustrated in Figs. 9.16 and 9.17. Figure 9.16 shows a projection onto the xz-plane of the ROV trajectory at the different force commands. As in previous cases, Fig. 9.17 further shows the behaviour of linear velocity components u, v, and w and angular velocity components p, q, and r.

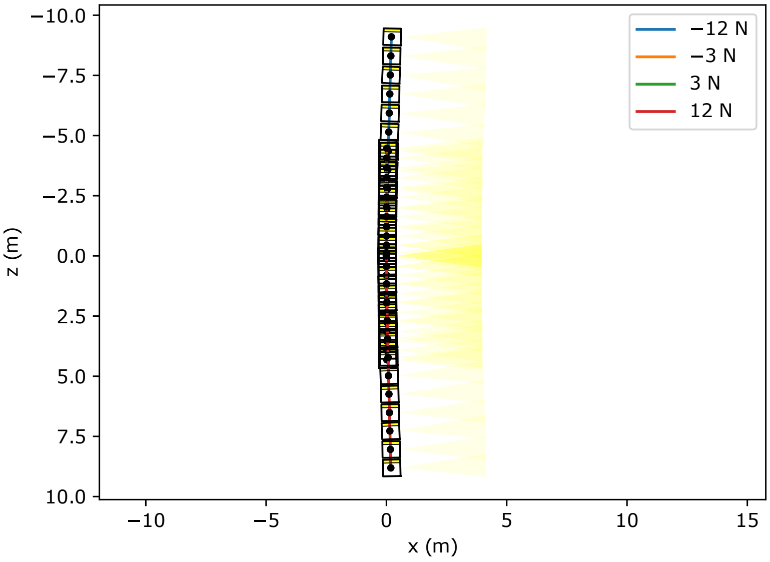


Figure 9.16. Vertical thrust at calm water and different heave commands: trajectories

Similarly as in the surge-direction case, the main difference between manoeuvring with and without thrust allocation resides in equating downward and upward actions. This can be checked in Fig. 9.16 by noticing that the ROV advances the same distance downwards and upwards. More precisely, in this simulation a downward heave command of 3 N assigns a propeller speed of 474 rpm to the vertical thruster; conversely, an upward command of -3 N assigns a propeller speed of -686 rpm. Moreover, at a greater force command, a downward command of 12 N assigns a propeller speed of 948 rpm and the corresponding upward command a propeller speed of -1373 rpm.

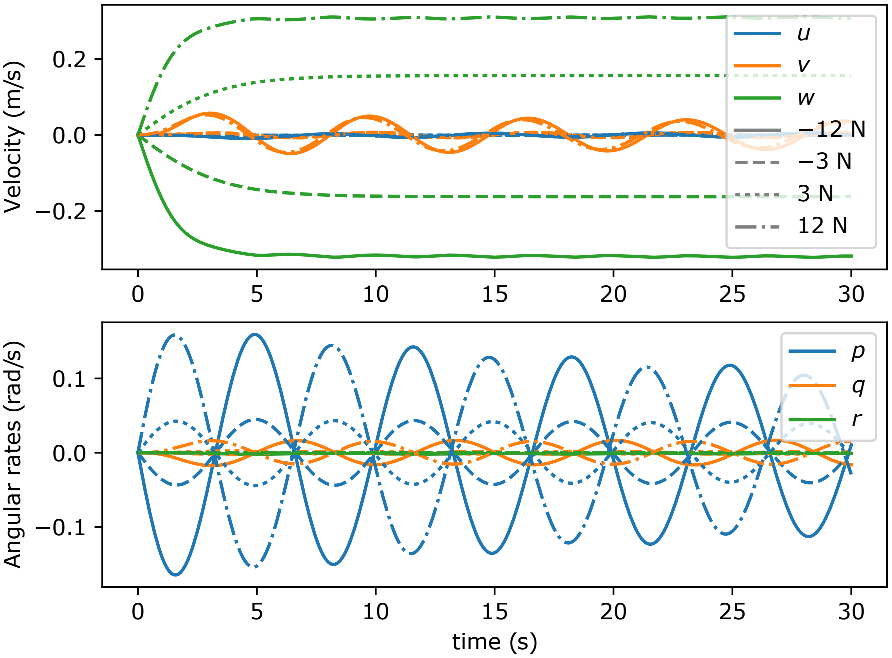


Figure 9.17. Vertical thrust at calm water and different heave commands: kinematic variables

From Figs. 9.16 and 9.17, it becomes apparent that the features of the ROV's motion are similar to those of the non-thrust-allocation case: the ROV is able to keep its trajectory approximately straight when not perturbed by current and the other degrees of freedom get excited as usual. These results suggest that attention to roll motion ought to be taken into account when controlling heave.

9.2.1.4 Yaw command

To induce yaw motion, a virtual force command is assigned to yaw input, namely for commands equivalent to moments of -3, -0.75, 0.75, and $e3$ Nm to allow clockwise and counterclockwise motion. The simulations are run for 30 s and the results are illustrated in Fig. 9.18.

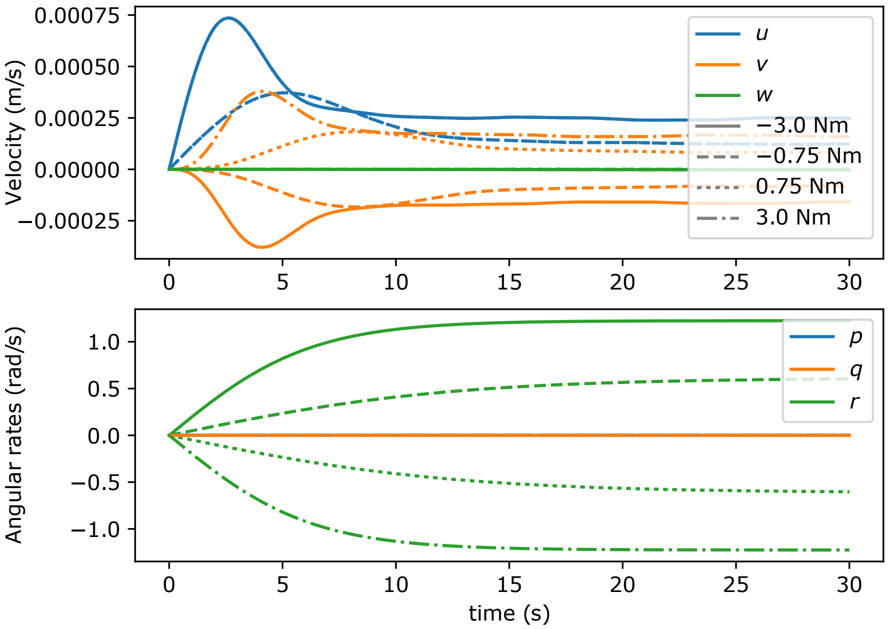


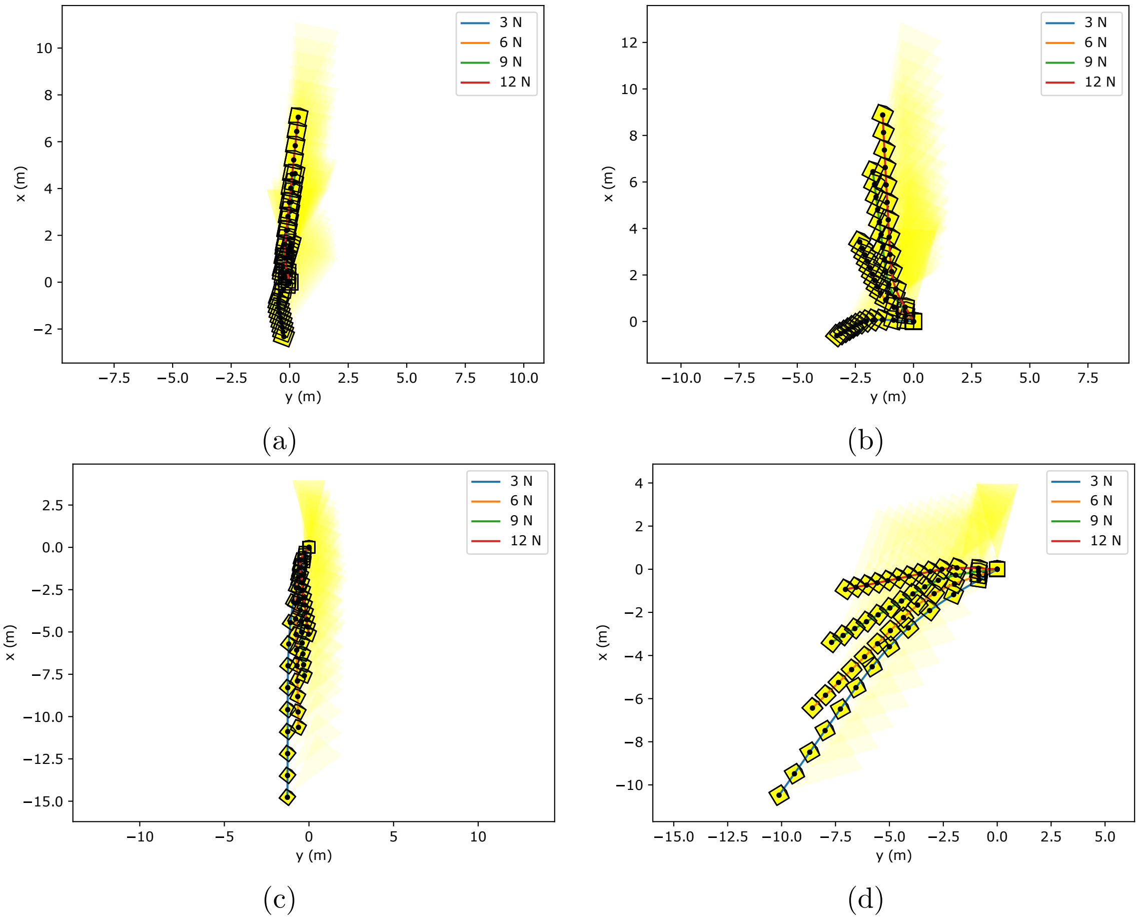
Figure 9.18. Yaw-inducing thrust at calm water and different propeller speeds: kinematic variables

From the kinematic variables shown in Fig. 9.18 it becomes apparent that the thrust allocation algorithm attenuates the excitation of other degrees of freedom. For instance, residual displacements, evidenced by the order of magnitude of u, v, and w, are greatly reduced.

9.2.2 SURGE MOTION IN OBLIQUE CURRENT

The oblique-current manoeuvres considered here replicate those in Sec. 9.1.2. More specifically, this means that the current conditions are the same as in Sec. 9.1.2, but different thrust conditions are tested by assigning different surge force commands in forward direction, namely commands equivalent to forces of 3, 6, 9, and 12 N. Concordantly, the results from all the scenarios are grouped according to all combinations of current conditions and illustrated by Figs. 9.19 and 9.20.

Figure 9.19 shows an xy-plane projection of the ROV's trajectory in all cases. Figures 9.19 (a) and (b) show behaviour at 0.2 m/s current and Figs. 9.19 (c) and (d) at 0.4 m/s. These motion scenarios are fundamentally the same as without thrust allocation: the only difference in this case is that motion is commanded by a virtual force. Because of this, the results are essentially the same as without thrust allocation. The ROV is able to withstand an imposed current of 0.2 m/s for force commands starting around 6 N for both 15 and 45 deg current directions;



(a)

(b)

(c)

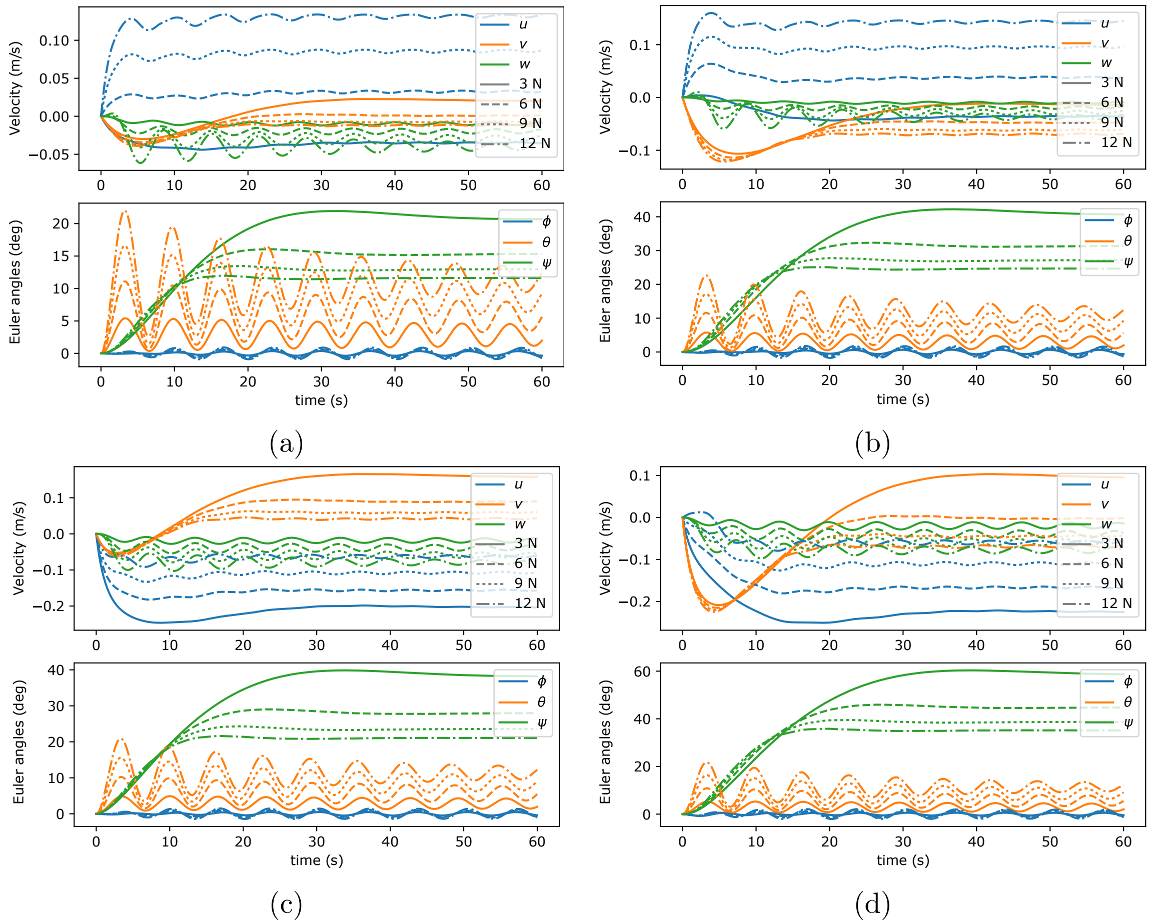
(d)

Figure 9.19. Surge motion in oblique-current using thrust allocation: trajectories

conversely, the ROV is not able to withstand an imposed current of 0.4 m/s in any case shown. In all cases is apparent how the ROV adopts certain heading related to the incoming current direction by reaching equilibrium around yaw. Further checking the results in Figs. 9.20 (a), (b), (c), and (d), increasing propeller speeds produces variations of pitch angle and lift force that induce vertical motion. This confirms the aforementioned difficulties of manoeuvring VISOR3 in currents that are not near-calm current, *i.e.*, starting around 0.4 m/s, and that the ROV is not easily manoeuvrable without further (feedback) control strategies.

9.3 PID CONTROL

The next step in studying VISOR3's manoeuvrability consists in simulating the ROV's response when PID feedback control loops are implemented. From previous results, the most immediate



(a)

(b)

(c)

(d)

Figure 9.20. Surge motion in oblique-current using thrust allocation: kinematic variables

control loops should be related to yaw and heave motion. For their simplicity, only simple, uncoupled PID control loops are considered here. More concretely, this means that to control yaw only heading and yaw rate are used as feedback and, conversely, to control heave only depth (or perhaps altitude) and heave velocity are used.

In the analyses to come, yaw control is considered first and heave control second. In these two cases, the control loops are heuristically tuned to an acceptable performance when subject to set-point changes, at zero current, and when the other degrees of freedom are not willingly disturbed. After the individual control loops are tuned, the response at certain motion scenarios is tested. First, the response to surge and sway is studied in calm water. Second, the response to surge is studied in oblique current.

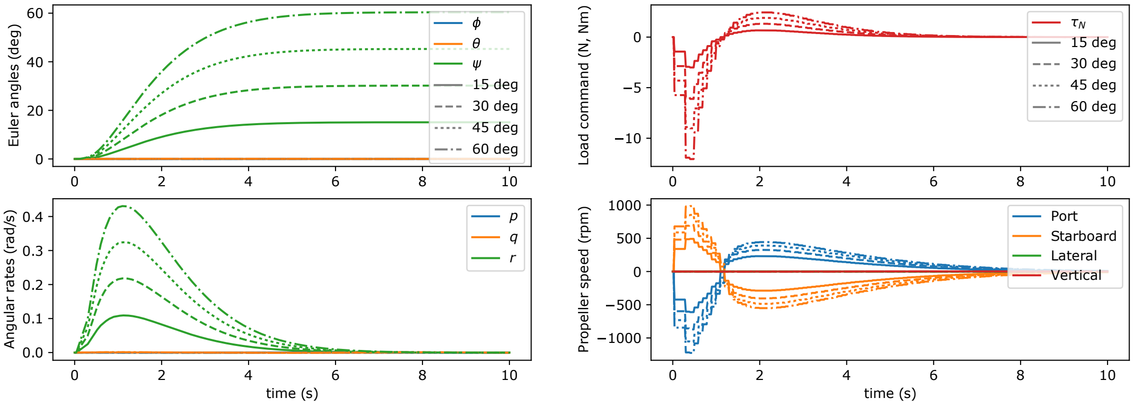
9.3.1 YAW CONTROL

As mentioned, the implemented yaw control law uses only heading and yaw rate as feedback, *i.e.*,

$$u_N = K_{P_\psi}(\psi_{\text{sp}} - \psi) - K_{D_r} r.$$

In the simulations, the tuning parameters, expressed in force and kinematic variables' units, are $K_{P_\psi} = 0.3$ Nm/deg and $K_{D_r} = 30$ Nm/(rad/s). This control law is implemented in discrete time with an update period (time step) of 0.1 s.

As shown in Fig. 9.21, the performance of the controller is verified by computing changes to heading set-point, namely from 0 to 15, 30, 45, and 60 deg. The figure shows the response of Euler angles, angular velocity components, yaw control command, and propellers speed. As shown, the controller was tuned to a response time of 5–6 s, without overshoot in heading, and without saturation in the actuators.



(a) (b)

Figure 9.21. Yaw control response

9.3.2 HEAVE CONTROL

The implemented heave control law uses only vertical coordinate and heave rate as feedback, *i.e.*,

$$u_Z = K_{P_z}(z_{\text{sp}} - z) - K_{D_w} w.$$

In the simulations, the tuning parameters, expressed in force and kinematic variables' units, are $K_{P_z} = 20$ N/m and $K_{D_w} = 50$ N/(m/s). As before, this control law is implemented in discrete time with an update period (time step) of 0.1 s.

The performance of the controller is verified by computing changes to vertical coordinate set-point, namely from 0 to −2, −1, 1, and 2 m. For these cases, Fig. 9.22 (a) shows the response of vertical coordinate and velocity components; conversely, Fig. 9.22 (b) show the response of Euler angles and angular velocity components. Figure 9.23 shows control effort by plotting the response of heave control command and propellers speed. As shown, the controller was tuned to a response time of 10–12 s, without overshoot in vertical coordinate. From Fig. 9.22 (b), it becomes noticeable how roll motion gets excited at magnitudes of roll angle in the order of 20 deg.

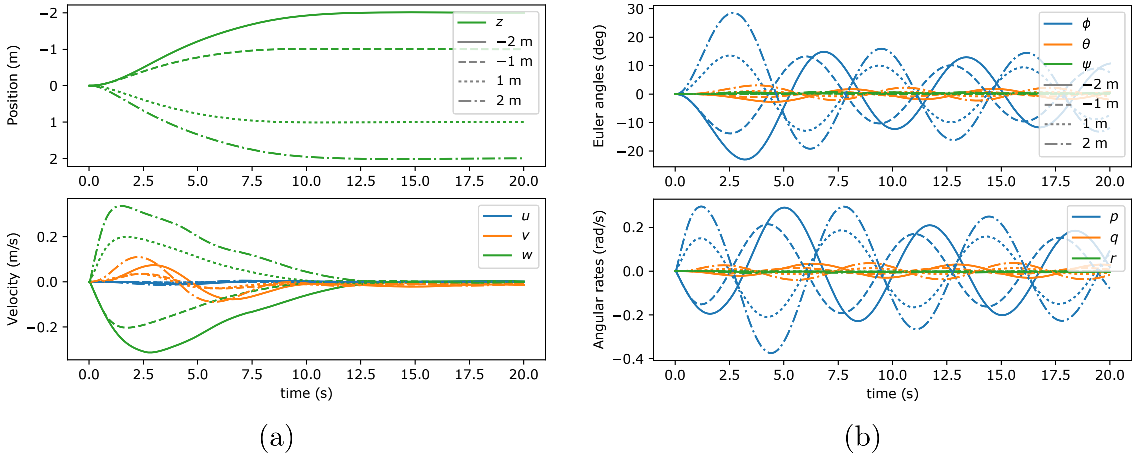


Figure 9.22. Heave control response: kinematics

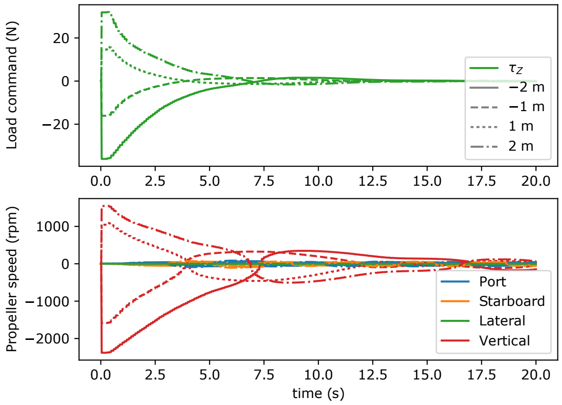
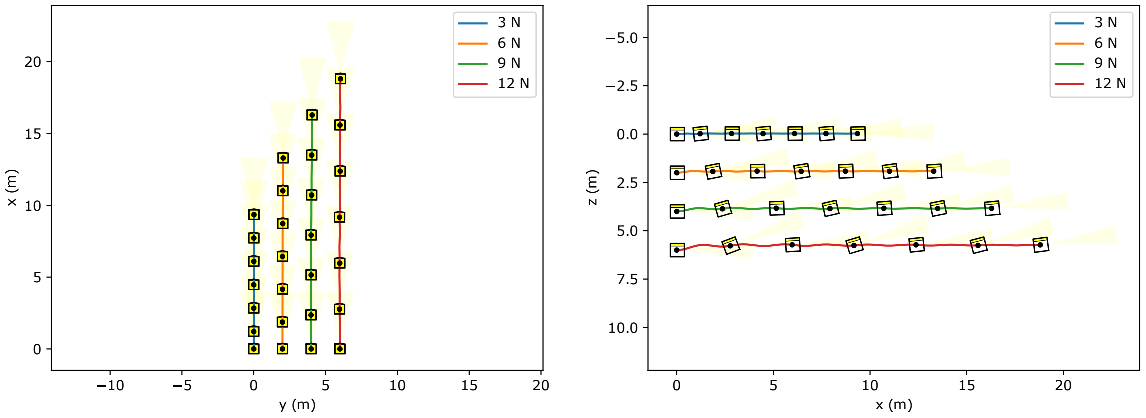


Figure 9.23. Heave control response: actuators

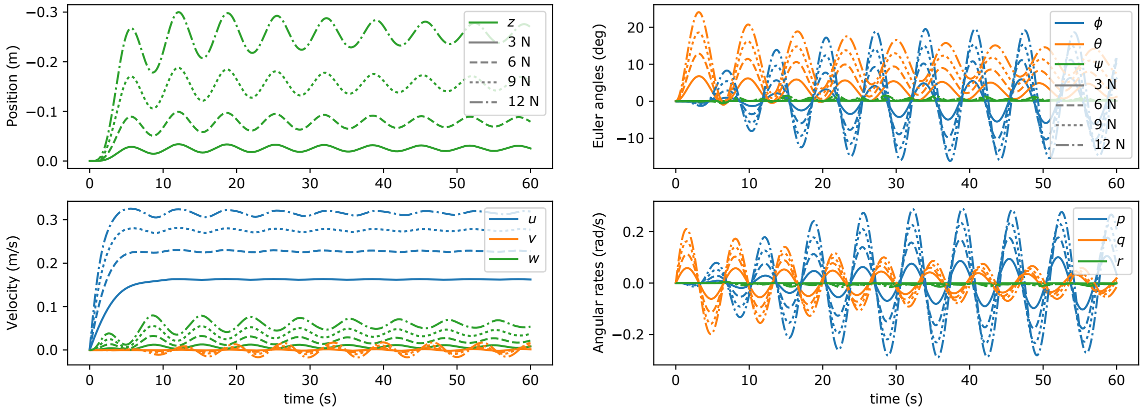
9.3.3 RESPONSE TO SURGE-DIRECTION COMMANDS

Surge-direction motion is performed as in previous open-loop cases, but in this case aforementioned yaw and heave controllers are active. To test the control system performance, a virtual force command is assigned to surge input, namely for forward motion commands equivalent to forces of 3, 6, 9, and 12 N. The simulations are run for 60 s and the results are illustrated in Figs. 9.24, 9.25, and 9.26. As usual, Figs. 9.24 (a) and (b) show the ROV trajectory at the different surge commands: Fig. 9.24 (a) shows the projection onto the xy-plane and Fig. 9.24 (b) the corresponding projection onto the xz-plane. Figure 9.25 further shows the behaviour of vertical coordinate, linear velocity components, Euler angles, and angular velocity components, and Fig. 9.26 illustrates control effort by plotting the response of heave control command and propellers speed.



(a) (b)

Figure 9.24. Surge motion response: trajectories



(a) (b)

Figure 9.25. Surge motion response: kinematics

As it becomes noticeable in Fig. 9.24 and evident in Fig. 9.25, after implementation of the heave control loop, the ROV is capable of keeping approximate horizontal motion. By using heave control, the vertical thruster is active and works to compensate the lift effect: this is apparent in Fig. 9.26. In addition, steady-state error in vertical coordinate is present, this is due, among other reasons, because the heave controller does not implement integral action. Nevertheless, an important dynamic effect that is not compensated is excitation of roll and pitch; from Fig. 9.25, one can notice that they are highly oscillatory.

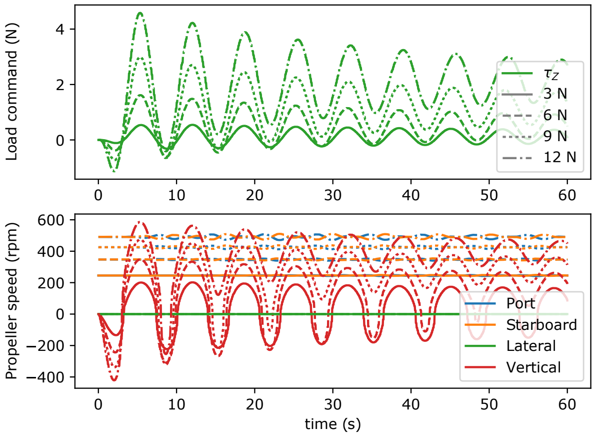


Figure 9.26. Surge motion response: actuators

9.3.4 RESPONSE TO SWAY-DIRECTION COMMANDS

Sway-direction motion is performed similarly as in the previous surge case, *i.e.*, to test the control system performance, a virtual force command is assigned to sway input, namely for forward motion commands equivalent to forces of 2, 4, 6, and 8 N. The simulations are run for 60 s and the results are illustrated in Figs. 9.27, 9.28, and 9.29. Figure 9.27 shows the xy-plane projection of the ROV trajectory at the different surge commands. Figure 9.28 further shows the behaviour of linear velocity components u, v, and w and angular velocity components p, q, and r, Fig. 9.29 (a) illustrates control effort by plotting the response of heave and yaw control commands, and Fig. 9.29 (b) illustrates propellers speed.

The importance of a yaw control loop for sway motion was acknowledged in previous analyses. From Fig. 9.27, it becomes apparent that after implementing the yaw control loop, the ROV is capable of keeping approximate lateral motion that gets deviated with increasing speeds. Moreover, as Fig. 9.29 shows, there is a control action related to yaw that increases with sway speed.

Because the heave control loop works as well, the vertical thruster compensates to assuage the lift effect: this is apparent in Fig. 9.29. Rather similarly as in the surge case, this type of motion excites roll; from Fig. 9.28, one can notice its highly oscillatory behaviour. Compensation of roll requires to be addressed to improve the control loop's performance.

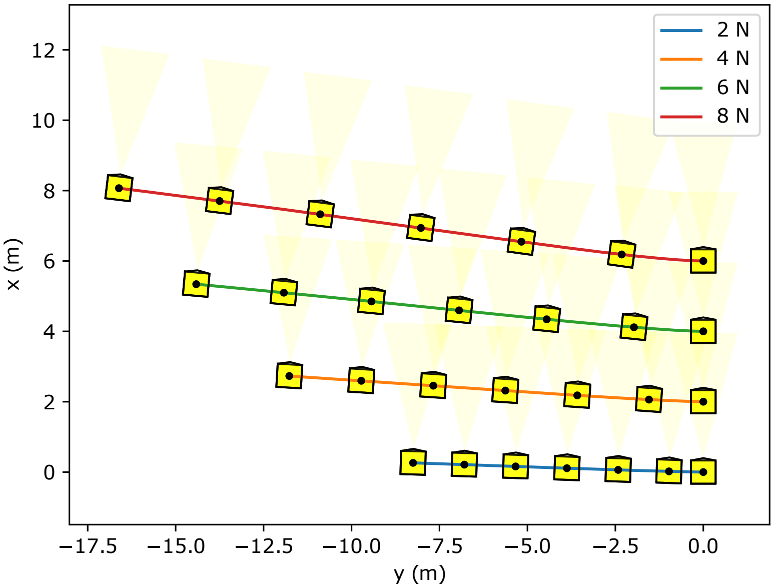
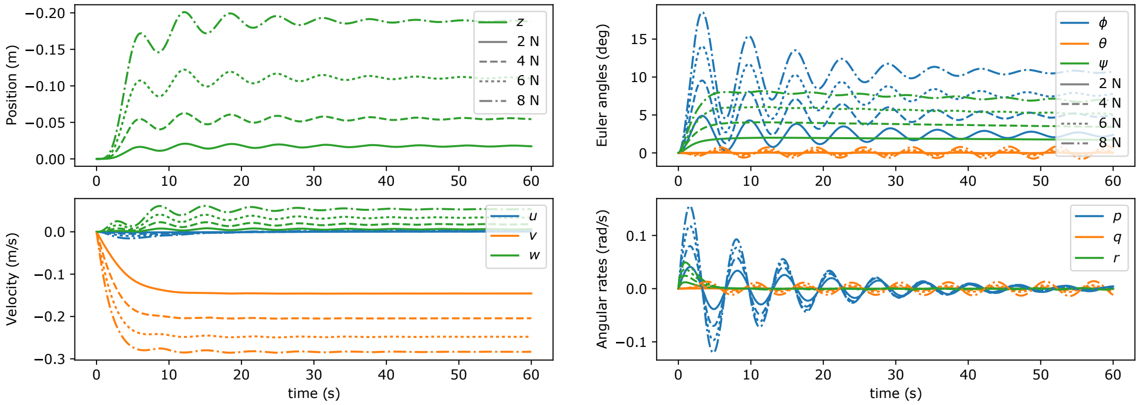


Figure 9.27. Lateral thrust at calm water and different sway commands: trajectories



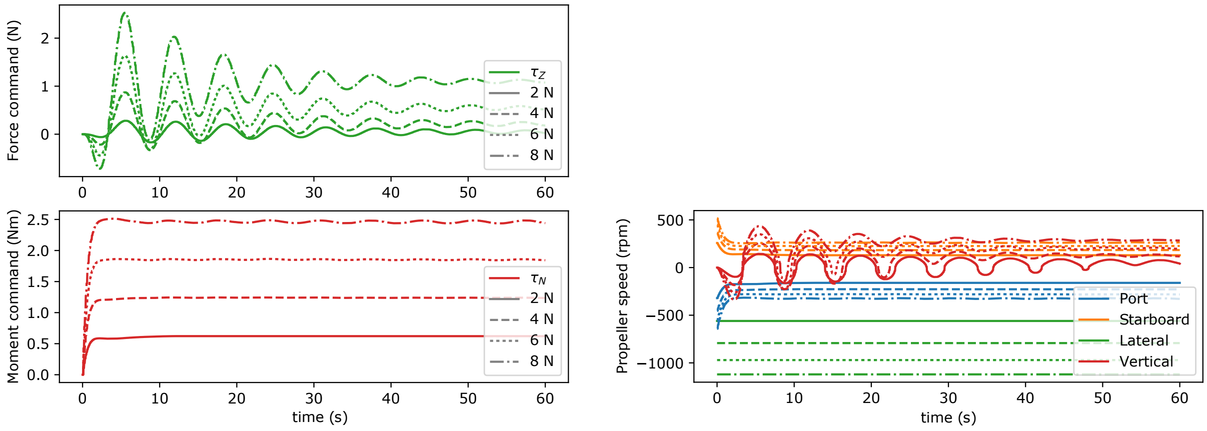
(a)

(b)

Figure 9.28. Sway motion response: kinematics

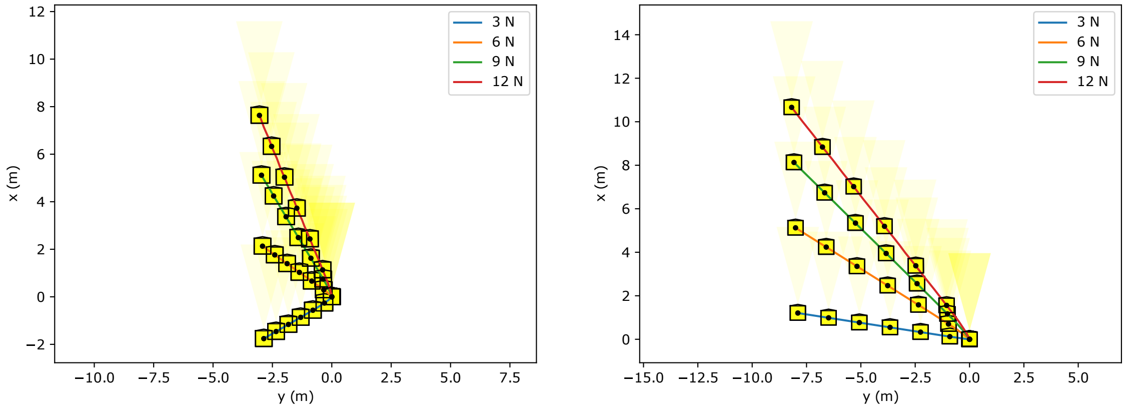
9.3.5 RESPONSE IN OBLIQUE CURRENT

The oblique-current manoeuvres considered here replicate those in Sec. 9.1.2, but in this case heave and yaw controllers are active and only 0.2 m/s current magnitude is considered. The current angles are the same, *i.e.*, 15 and 45 deg. Different thrust conditions are tested by assigning different surge force commands in forward direction, namely commands equivalent to forces of 3, 6, 9, and 12 N. Concordantly, the results from all the scenarios are grouped according to the two different current angles and illustrated by Fig. 9.30.



(a) (b)

Figure 9.29. Sway motion response: commands



(a) (b)

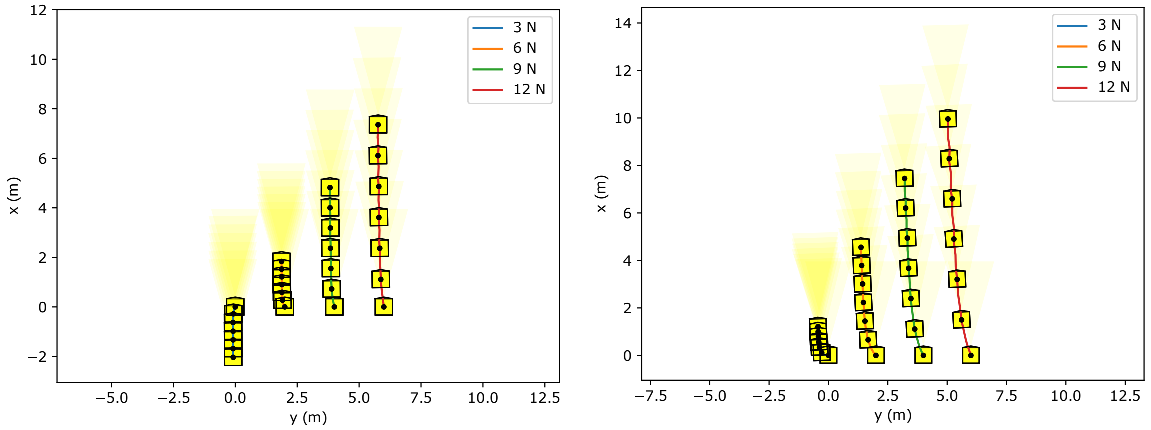
Figure 9.30. Surge motion in oblique-current using heave-yaw PID control: trajectories

Figure 9.30 shows an xy-plane projection of the ROV's trajectory in all cases. Figures 9.30 (a) and (b) show behaviour at 15 and 45 deg respectively. From the figures, the effect of yaw control is noticeable, because the ROV keeps heading approximately constant in all cases. Nevertheless, the fact that sway is not compensated is evident because the ROV drifts laterally. This effect can be addressed by adding a control loop to sway motion.

To reduce lateral drift, a simple sway motion PID loop may be given by the control law

$$u_Y = -K_{P_v} v - K_{I_v} \int_0^t v \mathrm{d}t.$$

In the simulations, the tuning parameters, expressed in force and kinematic variables' units, are $K_{P_v} = 40$ N/(m/s) and $K_{I_v} = 10$ N/m. As before, when this control law is implemented in discrete time, a time step of 0.1 s is used. The results of implementing a sway control loop are shown in Fig. 9.31. From observing the trajectories, the way lateral drift is compensated, compared to the trajectories in Fig. 9.30, becomes apparent. This demonstrates the pertinence of controlling sway, in addition to have and yaw, in the presence of disturbances such as current.



(a) (b)

Figure 9.31. Surge motion in oblique-current using sway-heave-yaw PID control: trajectories

9.4 CONCLUDING REMARKS

This chapter focused on performing time-domain simulations using VISOR3's time-domain model, to study its manoeuvrability. Overall, three groups of motion simulations were pursued: open-loop without thrust allocation, open-loop with thrust allocation, and multi-loop PID control. Additionally, in all groups, both calm-water and oblique-current manoeuvres were considered. Calm-water manoeuvres studied response of all controllable degrees of freedom (surge, sway, heave, and yaw), as well as circular motion where some degrees of freedom are combined. Oblique-current manoeuvres studied surge motion only. Closed-loop manoeuvres studied multi-loop PID schemes that focus initially on heave and yaw control and lastly on surge control. In this last case, the controllers are heuristically tuned to have acceptable performance at changes in set-point. Then, the closed-loop response to commands in surge and sway directions was studied first; conversely, the response to surge commands in oblique current without and with sway control was studied second.

By and large, the difference between motion scenarios without and with thrust allocation is that operator commands are simplified and thrust is equated between forward and backward motion. For the most part, the manoeuvring features in both cases are similar. Conversely and expectedly, the dynamic response of the ROV is considerably affected by closed-loop control. By using simple yaw and heave PID control, depth and heading are regulated when moving around other degrees of freedom. According to the results, there is considerable room for improvement for the control system response and robustness. Likewise, unwanted motion around non-controlled degrees of freedom, *i.e.*, roll and pitch oscillation, are left unaddressed.

As expected, this chapter's work leaves space for copious directions of advancement. First, a possible direction could be related to testing more complicated control schemes. Second, another direction could be the validation of the manoeuvring features in the actual ROV. Third, as it could be noted, integration with the cable model was not addressed as it is not trivial. And last, these manoeuvring studies could be pursued for different ROVs, such as Pionero500.

Chapter 10

Motion feasibility framework

This chapter proposes a framework for analysing motion feasibility in ROVs by means of numerical computations of the dynamics equations. Motion feasibility analyses are understood here as those that aim at answering questions about whether an ROV is able to move under certain operation conditions. Because in general that aim is unmanageably broad, a framework for analysing motion feasibility requires acknowledging the following aspects:

- An ROV integrates different components that affect whether an operation is feasible or not and the influence of each component varies in importance.
- Operation-scenario considerations may vary from simple to complex.
- Motion feasibility is relevant at different moments in the ROV's life cycle, for instance in design, testing, and operation planning/execution to name a few. Subsequently, what is known about the system varies at each stage, and what one wants to extract from the analysis varies as well.

Taking these ideas into consideration, a motion feasibility analysis framework should have the following desirable properties:

- The analysis should be possible to be carried through even when only basic information is available; consequently, what basic information means should be stated.
- The analysis should be able to improve its accuracy when more detailed information is added.
- The analysis should be able to be presented by means that are likely to be straightforwardly understood by people whose work/training is ROV-related.

By using the ideas stated above, the motion feasibility framework proposed in this work is stated in terms of the concept of ROV Dynamic Positioning Capability (ROV-DPCap): this chapter's objective is to show how this is possible.

10.1 DYNAMIC POSITIONING CAPABILITY IN ROVS

ROV Dynamic Positioning Capability (ROV-DPCap) analyses aim at determining whether an ROV motion system has enough actuator capacity to withstand environmental and operation-related loads, while keeping a prescribed position or motion. Moreover, it is understood that an ROV's Dynamic Positioning (DP) system is responsible for distributing power to the actuators in order to keep a constant position and heading or attain a prescribed motion. In this context, the DP system often includes, as subsystems, power, control (computer, sensors, position reference systems, operator panels, etc.), and actuator components. For an ROV, a DP system normally includes a human in the control loop, but it may function as well autonomously in some cases.

Because the concept of DPCap was originally developed for ships and surface vessels, these are used as a base to develop the DPCap framework for ROVs. The approach will be to extend the DNVGL-ST-0111 standard [34] from surface vessels to ROVs. Accordingly, this chapter proposes how to apply DPCap analyses to ROVs including the possibility of quasi-static and dynamic studies, as well as considering generic site-independent and site-specific conditions. Advances towards this goal were reported in [101]. To list some elements of ROV-DPCap, ocean current is considered as the primary environmental variable to determine capability, 2D and 3D plots are required to illustrate three-dimensional motion capability and account for three-dimensional relative current/motion, and, finally, the effects of vertical current profile on the cable and other depth- and operation-related loads might be included as well.

ROV-DPCap may be used to pursue offline analyses and serve as an online advisory/supervisory tool as well. Offline analyses are those that can be performed before an operation: this ranges from the design of a new ROV to planning an operation with an existing ROV. Possible analyses may be choosing a thruster configuration, computing whether a system is deployable into an specific location, *e.g.* in the Caribbean, among others. Conversely, as an online supervisory system, DPCap can be used to advise whether during an operation a loss of motion capability occurs, as well as to provide real-time operation decision-making information, such as estimating the conditions that may constitute risk or operation impairment. This thesis does not consider analyses or implementations of online advisory/supervisory tools.

10.2 METHODOLOGY OVERVIEW

10.2.1 ROV-DPCAP NUMBER DEFINITION

ROV-DPCap number is a way to indicate, in a rather simple manner, what type of environmental conditions an ROV is potentially capable to withstand. In this methodology, for an ROV, current is considered the most important type among all environment load-inducing conditions. For this reason, the ROV-DPCap number is quantified from incoming current. Table 10.1 shows the relationship between incoming current and ROV-DPCap number. As the table shows, ROV-DPCap number is defined by keeping a subjective parallel to Beaufort (BF) scale's numbers and definitions [34]. Also, the different ROV-DPCap numbers are related to a current value in m/s, *i.e.*, ROV-DPCap numbers from 0 to 11 represent currents starting from 0 to 2.2 m/s, by using 0.2 m/s steps.

Table 10.1. DPCap numbers

ROV-DPCap number	Beaufort (BF) number	Beaufort description	Current speed [m/s]
0	0	Calm	0.0
1	1	Light air	0.2
2	2	Light breeze	0.4
3	3	Gentle breeze	0.6
4	4	Moderate breeze	0.8
5	5	Fresh breeze	1.0
6	6	Strong breeze	1.2
7	7	Moderate gale	1.4
8	8	Gale	1.6
9	9	Strong gale	1.8
10	10	Storm	2.0
11	11	Violent storm	2.2
12	NA	Hurricane force	NA

This considered, a certain ROV-DPCap number indicates that *the ROV can keep its position at the corresponding ROV-DPCap number conditions and all conditions below, but not in the conditions specified for the next DPCap number.* This means, for instance, that if the ROV-DPCap number is 5 then the ROV can withstand a current of 1.0 m/s but not a current of 1.2 m/s; furthermore, the exact limiting current should be somewhere in between 1.0 and 1.2 m/s.

10.2.2 TYPES OF ANALYSES

The different types of ROV-DPCap analyses are obtained by using either steady-state, namely pseudo-static, or dynamic balances and considering either standard or site-dependent conditions. Firstly, a *pseudo-static* balance is a balance of forces and moments in steady-state conditions, but external loads are augmented by using a factor of safety (*e.g.*, 1.25) to account for dynamic effects and non-modelled loads. A *dynamic* balance uses, *e.g.*, Newton's second law or equivalent to consider transient phenomena such as the effect of time-dependent environmental conditions, inertia, and control-system dynamics. Secondly, *standard conditions* refers to analyses that allow one to potentially compare different ROVs as fairly as possible. Consequently, in this framework, *standard conditions* refers to considering current-induced loads by using the values stated in Table 10.1 and testing different relative current directions. *Site-dependent conditions* refers to adding any other environment-induced or operation-related loads particular to each operation scenario, *e.g.*, cable-, depth-, and wave-related loads. Roughly speaking, standard conditions compare ROVs; site conditions assess operations.

The four types of ROV-DPCap analyses proposed are L2, L2s, L3, and L3s; this is done to keep a parallel to the DNVGL-ST-0111 standard [34]. As mentioned, the different types of analyses cover either quasi-static or dynamic conditions and either standard or site-specific conditions; the name of each type of analysis is as follows:

- L2: quasi-static, standard conditions.
- L2s: quasi-static, site-specific conditions.
- L3: dynamic, standard conditions
- L3s: dynamic, site-specific conditions.

Table 10.2 summarises the aforementioned analysis classification.

Table 10.2. DPCap analyses

		Balance type	
		Quasy-static	Dynamic
Load conditions	Standard	L2	L3
	Site-specific	L2s	L3s

Worth mentioning, the DNVGL-ST-0111 standard proposes as well a Level 1 (L1) analysis, intended only for ship-shaped mono-hull vessels. Such type of analysis does not translate to ROVs directly; thus, it is not considered here.

10.2.3 ROV-DPCAP PLOTS

ROV-DPCap is illustrated by using DPCap plots. DPCap plots were born as polar graphs that show the operational envelope of a vessel. They depict the limiting environmental conditions a vessel can withstand in DP mode, at each direction (or at least a set of directions) in the xy-plane and at normal and failure conditions. For ROVs, two types or plots are proposed for presenting DPCap: 2D and 3D. In general, NED frame is used to draw these plots (x-direction is considered positive from aft to bow, y-direction positive from port to starboard, and z-direction positive down) following the convention in [48].

Two-dimensional plots are polar plots that show results for DP capability on the xy, xz, and yz planes, *i.e.*, considering that motion occurs only on the specified plane. These options are depicted in Fig. 10.1. The polar angle and radius represent relative velocity direction and ROV-DPCap number respectively. From the point of view of eq. (2.4), *i.e.*,

$$\nu_r = \begin{bmatrix} u_r & v_r & w_r \end{bmatrix}^\mathsf{T} = \begin{bmatrix} V_r\cos(\alpha_r)\cos(\beta_r) & V_r\sin(\beta_r) & V_r\sin(\alpha_r)\cos(\beta_r) \end{bmatrix}^\mathsf{T},$$

each 2D plot is calculated by using the following considerations:

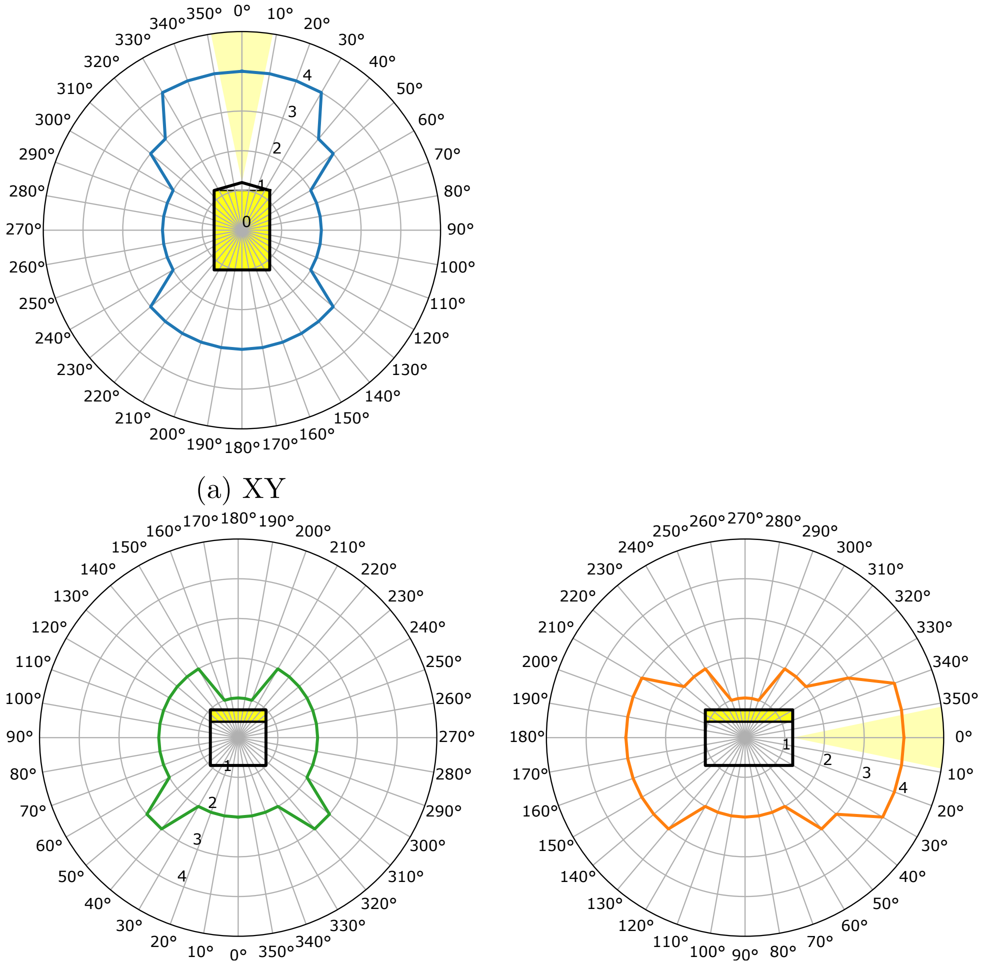
- xy-plane plot: $\alpha_r = 0$ and $0 \leq \beta_r < 360$ deg, with 10 deg increments.
- xz-plane plot: $\beta_r = 0$ and $\beta = 90$ deg, and $0 \leq \alpha_r < 180$ deg, with 10 deg increments.
- yz-plane plot: $\alpha_r = 90$ deg and $0 \leq \beta_r < 360$ deg, with 10 deg increments.

Three dimensional plots show ROV-DPCap around all directions, *i.e.*, from the point of view of eq. (2.4), $0 \leq \alpha < 180$ deg and $0 \leq \beta < 360$ deg. As illustrated in Fig. 10.1, 3D plots are spherical plots where, similarly as in 2D plots, orientation and radius represent relative velocity direction and ROV-DPCap number respectively. Please note how, in a red-white-blue scale, colours show variation of ROV-DPCap number from 1 to 5. Red indicates the lowest and blue the best capability. In this case, heave and sway motion are more restricted that surge's.

Recalling results from Chapter 4, geometrically, relative velocity could be represented by a point in $\mathbb{S}^2$ (a two-dimensional sphere in $\mathbb{R}^3$) and the task of distributing points uniformly therein is not trivial. For instance, the plot in Fig. 10.2 was computed by using a mapping $\mathrm{S}^2_s : \mathbb{R}^2 \to \mathbb{S}^2$, given by eq. (2.4) as

$$(\alpha_r, \beta_r) \mapsto (e_u, e_v, e_w) = (\cos(\alpha_r)\cos(\beta_r), \sin(\beta_r), \sin(\alpha_r)\cos(\beta_r)),$$

i.e., using an Euler angles parametrisation. From Fig. 10.2, it becomes apparent that some computation points regions are sparser that others. Here, fairer distributions of points on the sphere will be useful for arranging DPCap computations and plots; consequently, further results will require using concepts from design of computer experiments (see Chapter 5).



(a) XY

(c) YZ

(b) XZ

Figure 10.1. 2D ROV-DPcap plots example

10.2.4 ROV-DPCAP ANALYSIS PROCEDURE

The process of computing an ROV-DPCap analysis is illustrated here by assuming standard conditions, *i.e.*, L2 and L3 analyses. The overall process follows three steps, namely

1. Pre-process: create a design table that specifies all the directions where the ROV-DPCap number is computed. The result of this step is a design table with a list of unitary vectors in $\mathbb{R}^3$, *i.e.*, a list of points $\mathbf{e}_{\nu_i} \in \mathbb{S}^2$, for $i = 1, \dots, N$.
2. Process: compute the limiting conditions and ROV-DPCap number for each point $\mathbf{e}_{\nu_i}$ from the design table.
3. Post-process: create ROV-DPCap plots, auxiliary plots, and report results.

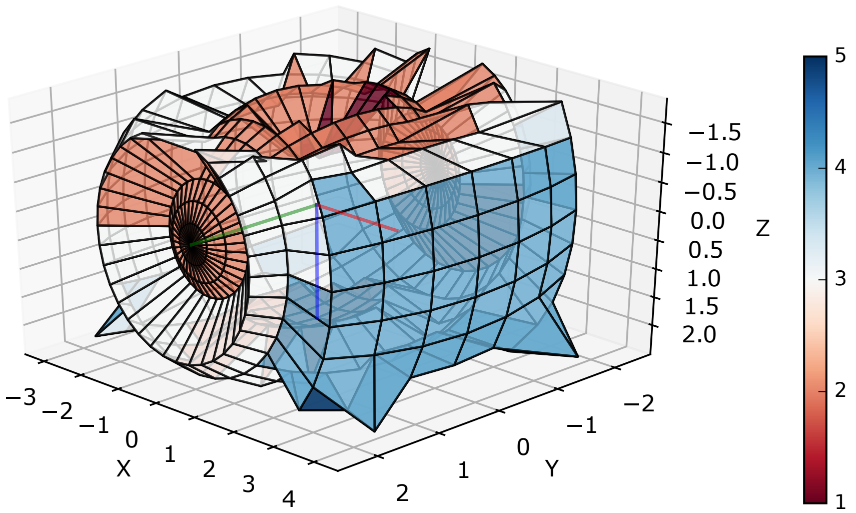
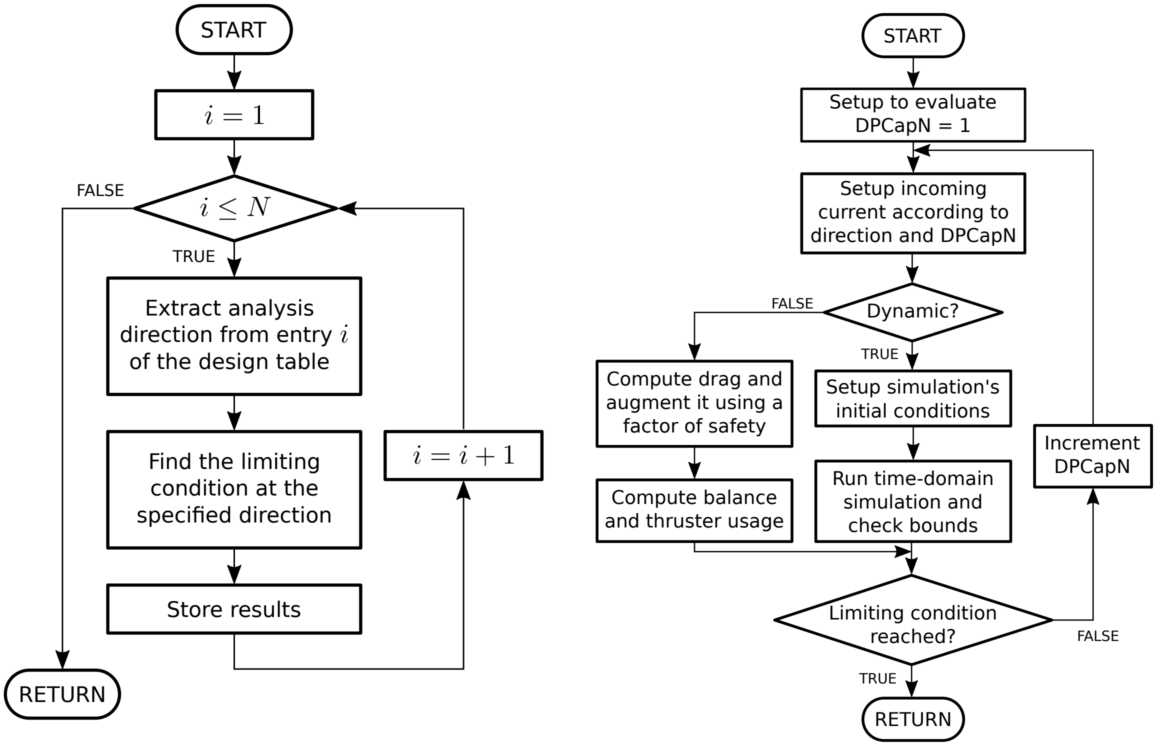


Figure 10.2. 3D ROV-DPcap plots example

Preprocess step requires defining a design table that follows similar guidelines as shown in Chapter 5. For ROV-DPCap three types of designs are proposed here:

1. 1D: these designs are parametrised in terms of one parameter. These are designs that geometrically resemble points on a circle (line) in $\mathbb{S}^2$, so their parameter is an angle. More specifically, these are mainly designs where, *e.g.*, only xy, xz, and yz plane points are considered. This type of designs is considered in Sec. 4.1.1.
2. 2D: these designs are parametrised in terms of two parameters. These are designs that fill the $\mathbb{S}^2$ space by using a two dimensional parametrisation such as S^2_s and S^2_c. In general, these parametrisations are defined as $(q_1, q_2) \mapsto (e_u, e_v, e_w)$, as specified in Sec. 4.1.2. This makes it possible to create a full factorial design (lattice or grid) in $\mathbb{R}^2$ and translate it to $\mathbb{S}^2$, similar to those shown in Figs. 4.1 and 4.2. This type of designs was used to create the ROV-DPCap plot in Fig. 10.2.
3. 3D: these designs are obtained by distributing points directly in $\mathbb{S}^2$, by using criteria such as covering or packing. These designs where discussed in Sec. 4.1.3. This type of designs is illustrated in Fig. 4.3.

Process step is the core of ROV-DPCap analysis. It requires to loop around all N elements of the design table and find the limiting condition that defines an ROV-DPCap number. This loop is illustrated in Fig. 10.3 (a). The way the limiting condition is computed depends on whether the analysis is L2, L2s, L3, or L3s. To illustrate the procedure, only standard conditions are assumed, *i.e.*, L2 and L3 for pseudo-static and dynamic cases respectively. An algorithm to compute the limiting condition is shown in Fig. 10.3 (b). It is important to note that the algorithm shown here is comprehensive; this means that all possible DPCap numbers


START
$i = 1$
FALSE
$i \leq N$
TRUE
Extract analysis direction from entry i of the design table
Find the limiting condition at the specified direction
Store results
$i = i + 1$
RETURN
START
Setup to evaluate DPCapN = 1
Setup incoming current according to direction and DPCapN
FALSE
Dynamic?
TRUE
Compute drag and augment it using a factor of safety
Compute balance and thruster usage
Setup simulation's initial conditions
Run time-domain simulation and check bounds
Increment DPCapN
Limiting condition reached?
FALSE
TRUE
RETURN

(a) (b)

Figure 10.3. ROV DPCap algorithm

are considered from 1 to the limiting condition. This is to say that a smarter algorithm may be implemented, so that fewer computations are required to find the limiting condition at each direction.

As illustrated in Fig. 10.3 (b), there is a relevant difference in the manner pseudo-static analyses are performed compared to dynamic ones. Pseudo-static analyses require to compute steady-state loads and augment them by using a factor of safety, usually 1.25. Then, the augmented loads are used to compute required thruster forces by performing a steady-state balance and using a thrust allocation scheme. Finally, the limiting condition is obtained when any of the actuators is required to give more than 100 % thrust. Conversely, dynamic analyses require to setup and run a time-domain simulation of the ROV's motion during a representative time-window. The limiting condition is obtained when, during motion, the ROV surpasses certain position and heading bounds. Using the results in [122] as reference, a ±0.2 m tolerance for position and a ±3 deg tolerance for heading may be proposed.

Post-process step requires consolidating stored results, creating ROV-DPCap and auxiliary plots, and reporting results. The overall results include:

- ROV-DPCap number at each condition.
- Thruster usage data and statistics.
- Time-domain motion variables for dynamic analyses and statistics.

10.3 ROV-DPCAP ANALYSES

This section shows how the considerations for each ROV-DPCap analysis type is taken into account in the ROV model. The approach is to illustrate how model (2.5), namely

$$\mathbf{M}_{RB}\dot{\nu} + \mathbf{C}_{RB}\left(\nu\right)\nu + \mathbf{M}_{A}\dot{\nu}_{r} + \mathbf{C}_{A}\left(\nu_{r}\right)\nu_{r} + \mathbf{D}\left(\nu_{r}\right)\nu_{r} + \mathbf{g}\left(\eta\right) = \tau(\mathbf{u}) + \tau_{\text{cable}} + \tau_{\text{wave}},$$

is modified in each case. In all cases, model (2.16) for thrust forces is required, namely

$$\tau = \mathbf{T}\mathbf{f}(\mathbf{u}).$$

This is specially important for L2 and L2s analyses, where special attention must be paid to the behaviour of $\mathbf{f}(\mathbf{u})$. Thrust allocation plays an important role too, because inverse map (2.20), *i.e.*

$$\mathbf{f}(\mathbf{u}) = \mathbf{T}^{\dagger}\mathbf{u},$$

defines how control signals $\mathbf{u}$ are translated into thruster forces.

10.3.1 L2 ANALYSIS

For L2 analyses, steady-state conditions are assumed, *i.e.*, $\dot{\nu} = \dot{\nu}_r = 0$. More specifically, the following considerations are taken:

- Zero speed, *i.e.*, $\nu = 0$.
- The vehicle is well-balanced and all attitude angles are zero at equilibrium, *i.e.*, $\phi = \theta = 0$. Furthermore, it can be assumed that $\psi = 0$ without loss of generality.
- Zero payload, *i.e.*, $m_p = 0$ in (2.8).
- Neutral buoyancy, *i.e.*, $W = B$. This makes the gravity vector equal to zero.
- Drag due to current is the only environmental load.
- All external loads are ignored.

This reduces model (2.5) to

$$\mathbf{D}\left(\nu_{r}\right)\nu_{r} = \tau,$$

i.e., thrusters compensate only for drag. Combining this with thrust net load and thrust allocation and using factor of safety F_s, the steady-state balance for L2 analysis becomes

$$F_s \mathbf{D}\left(\nu_r\right)\nu_r = \mathbf{T}\mathbf{f}(\mathbf{u}),$$

along with thrust allocation maps (2.20) and (2.22), *i.e.*, $\mathbf{f}_u = \mathbf{T}^{\dagger}\tau_u$ and $\mathbf{u} = \mathbf{f}_u^{\dagger}(\mathbf{f}_u)$. The thrust allocation algorithm is given by the sequence of maps that compute thruster commands from virtual force command, *i.e.*, $\tau_u \mapsto \mathbf{f}_u \mapsto \mathbf{u}$.

This considered, solving the pseudo-static balance problem requires finding virtual force command τ_u given ν_r. For this problem, the residual equation $\mathbf{r}(\tau_u) = \mathbf{0}$ becomes

$$\mathbf{r}(\tau_u) = F_s \mathbf{D}\left(\nu_r\right)\nu_r - \mathbf{T}\,\mathbf{f}\left(\mathbf{f}_u^{\dagger}\left(\mathbf{T}^{\dagger}\tau_u\right)\right) = \mathbf{0}.$$

This equation is to be solved with a root-finding algorithm. After a solution for each ν_r is found, a limiting condition is reached when for any thruster i the required propeller speed surpasses its maximum, *i.e.*, $n_i > n_{\max_i}$.

10.3.2 L2S ANALYSIS

For L2s analyses, steady-state conditions are assumed as well, but site-specific environmental conditions and loads are included. More specifically, the following considerations are made:

- Zero speed, *i.e.*, $\nu = 0$.
- The vehicle is well-balanced and all attitude angles are zero, *i.e.*, $\phi = \theta = 0$. Furthermore, it can be assumed that $\psi = 0$ without loss of generality.
- Payload is included when applicable.
- Buoyancy is not necessarily equal to weight. This may produce a steady gravity load, but, given that zero attitude is considered, restoring moments do not apply.
- All site-specific and depth-dependent environmental loads may be included. For instance, wave loads may be considered if operating at wave influence zone; consequently, the added mass coefficients will be frequency dependent as described in [43, 120], and the assumptions of (2.9) do not hold.
- External loads, as well as depth- and site-dependent loads are included; for instance, accumulated drag on the cable and closeness to bottom and underwater structures.

This reduces model (2.5) to

$$\mathbf{D}\left(\nu_r\right)\nu_r + \mathbf{g}\left(\eta\right) - \tau_{\text{cable}} - \tau_{\text{wave}} = \tau,$$

where $\mathbf{g}(\eta) = \begin{bmatrix} 0, & 0, & -(W-B), & 0, & 0, & 0 \end{bmatrix}^{\mathsf{T}}$. Here thrusters compensate for drag and other site-dependent loads. As it was done for L2 analysis, combining this with thrust net load and thrust allocation and using factor of safety F_s, the steady-state balance for L2s analysis becomes

$$F_s \left(\mathbf{D}\left(\nu_r\right)\nu_r + \mathbf{g}\left(\eta\right) - \tau_{\text{cable}} - \tau_{\text{wave}}\right) = \mathbf{T}\mathbf{f}(\mathbf{u}).$$

along with thrust allocation maps (2.20) and (2.22), *i.e.*, $\mathbf{f}_u = \mathbf{T}^{\dagger}\tau_u$ and $\mathbf{u} = \mathbf{f}_u^{\dagger}(\mathbf{f}_u)$. The pseudo-static balance problem requires finding virtual force command τ_u given ν_r. For this problem, the residual equation $\mathbf{r}(\tau_u) = \mathbf{0}$ becomes

$$\mathbf{r}(\tau_u) = F_s \left(\mathbf{D}\left(\nu_r\right)\nu_r + \mathbf{g}\left(\eta\right) - \tau_{\text{cable}} - \tau_{\text{wave}}\right) - \mathbf{T}\,\mathbf{f}\left(\mathbf{f}_u^{\dagger}\left(\mathbf{T}^{\dagger}\tau_u\right)\right) = \mathbf{0}.$$

As before, this equation is to be solved with a root-finding algorithm. After a solution for each ν_r is found, a limiting condition is reached when for any thruster i the required propeller speed surpasses its maximum, *i.e.*, $n_i > n_{\max_i}$. The computation of τ_{cable} is done as explained in Chapter 7.

10.3.3 L3 ANALYSIS

For L3 analyses, dynamic conditions are assumed, *i.e.*, $\dot{\nu} \neq 0$ and $\dot{\nu}_r \neq 0$. This type of analysis requires the following considerations:

- DP control is activated: position and heading.
- The vehicle is well-balanced and all attitude angles are zero at equilibrium, but ϕ, θ, and ψ are variable.
- Zero payload, *i.e.*, $m_p = 0$ in (2.8).
- Neutral buoyancy, *i.e.*, $W = B$. This makes the first three components of the gravity vector equal to zero but, in general, the restoring moments different to zero.
- Drag due to current is the only environmental load. Current may be time-variant.
- All external loads are ignored.

This reduces model (2.5) to

$$\mathbf{M}_{RB}\dot{\nu} + \mathbf{C}_{RB}\left(\nu\right)\nu + \mathbf{M}_A\dot{\nu}_r + \mathbf{C}_A\left(\nu_r\right)\nu_r + \mathbf{D}\left(\nu_r\right)\nu_r + \mathbf{g}\left(\eta\right) = \tau,$$

where

$$\mathbf{g}(\eta) = \begin{bmatrix} 0, & 0, & 0, & -(z_gW - z_bB)\cos\theta\sin\phi, & (z_gW - z_bB)\sin\theta, & 0 \end{bmatrix}^{\mathsf{T}}.$$

This (ODE) model has to be solved along with models (2.16) and (2.20) for thrust net load and thrust allocation, as well as with controller model (2.23) or equivalent. For L3 analyses, limiting conditions depend on position and heading bounds.

10.3.4 L3S ANALYSIS

For L3s analyses, dynamic conditions are assumed as well, but site-specific environmental conditions and loads are included. More specifically, the following considerations are made:

- DP control is activated: position and heading.
- The vehicle is well-balanced and all attitude angles are zero at equilibrium, but ϕ, θ, and ψ are variable.
- Payload is included when applicable.
- Buoyancy is not necessarily equal to weight. This makes that gravity forces as well as restoring moments are included.
- All site-specific and depth-dependent environmental loads may be included. For instance, wave loads may be considered if operating at wave influence zone; consequently, added mass coefficients will be frequency dependent [43, 120] and assumptions of (2.9) do not hold.
- External loads, as well as depth- and site-dependent loads are included; for instance, cable's accumulated drag and motion and closeness to bottom and underwater structures.

This makes that complete model (2.5) must be considered. This model has to be solved along with models (2.16) and (2.20) for thrust net load and thrust allocation, as well as with controller model (2.23) or equivalent. For L3s analyses, limiting conditions depend on position and heading bounds.

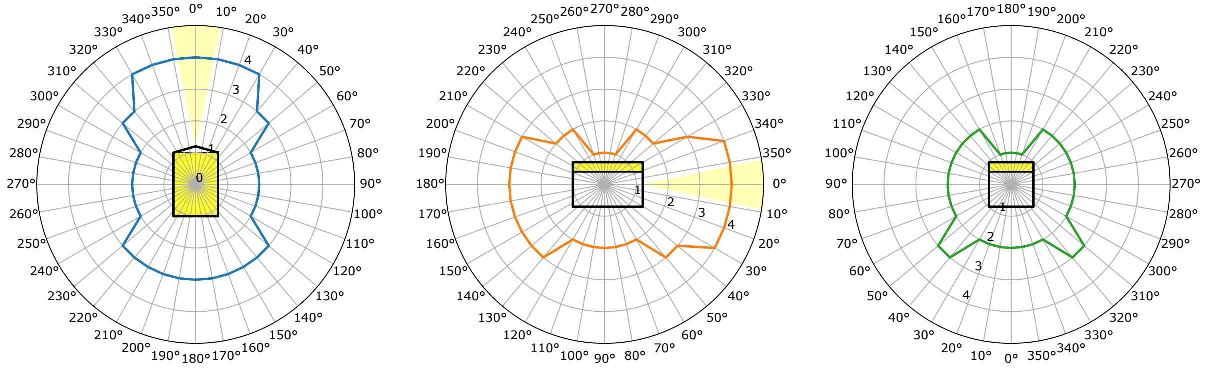
10.4 CASE STUDIES

In this section different case studies are considered to illustrate the most important features of ROV-DPCap, namely:

- L2 analysis of Minerva,
- L2 analysis of visor3,
- L2 analysis of two thruster configurations during ROV design, and
- L2s analysis where cable drag accumulation is considered.

10.4.1 L2 ANALYSIS OF MINERVA

According to Sec. 10.3.1, the L2 analysis on Minerva is done by computing a quasi-static balance of drag and thruster forces and using a thrust allocation algorithm. To compute the quasi-static balance, a factor of safety $F_s = 1.25$ is used. The Minerva's model and thrust allocation algorithm used herein are as described in Sec. 3.3. After computing the thrust allocation algorithm, the resulting two-dimensional plots of ROV-DPCap number are shown in Fig. 10.4; moreover, the three-dimensional plots are shown in Fig. 10.5.



(a) XY (b) XZ (c) YZ

Figure 10.4. 2D ROV-DPcap plots of Minerva

In Fig. 10.4 (a), the xy-plot illustrates the horizontal plane motion capability. From this plot, it is apparent that there is more capability for forward motion, with ROV-DPCap number around 4, than for backward motion, with ROV-DPCap number around 3: this due to the difference between forward and backward thrust of unidirectional thrusters. It is also apparent that there is more capability for overall surge motion than for sway motion, where ROV-DPCap number is around 2. Conversely, xz- and yz-plots illustrate capability for vertical motion: see Figs. 10.4 (b) and (c) respectively. For instance, the xz-plot shows vertical motion capability compared to surge motion.

10.4.2 L2 ANALYSIS OF VISOR3

The L2 analysis on visor3 is done similarly as Minerva's; consequently, the quasi-static balance of drag and thruster forces is computed by using a factor of safety $F_s = 1.25$. The visor3's model and thrust allocation algorithm used herein are as described in Sec. 3.1. As previously, after computing the L2 ROV-DPCap algorithm, the resulting two-dimensional plots of ROV-DPCap number are shown in Fig. 10.6 and the three-dimensional plots are

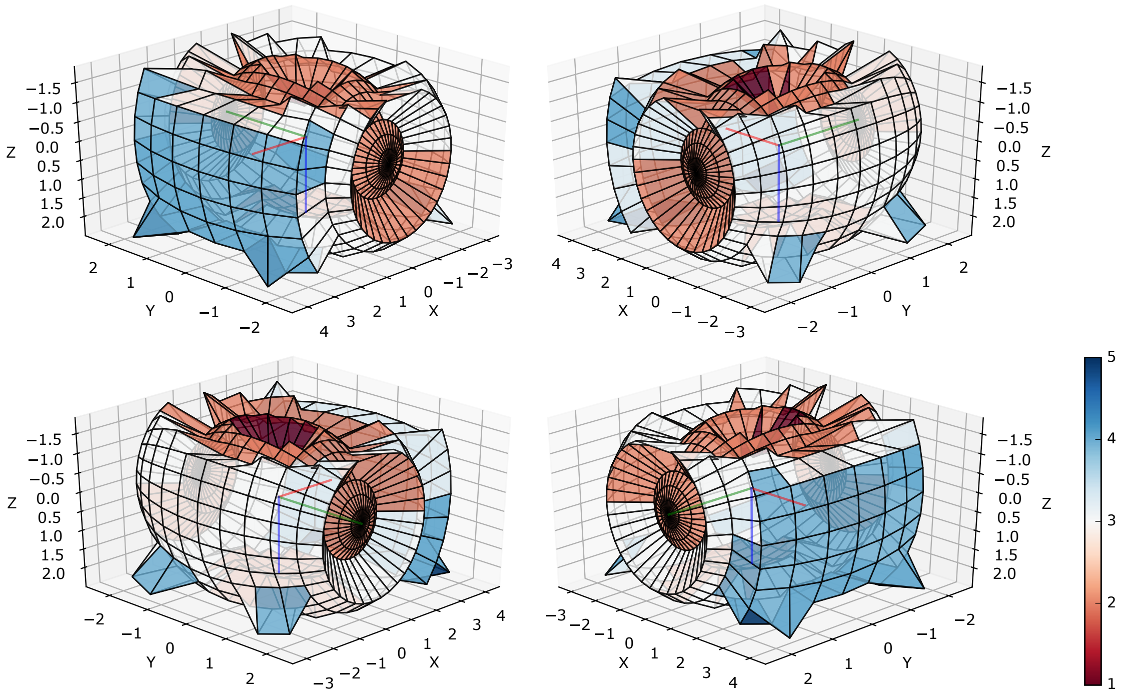
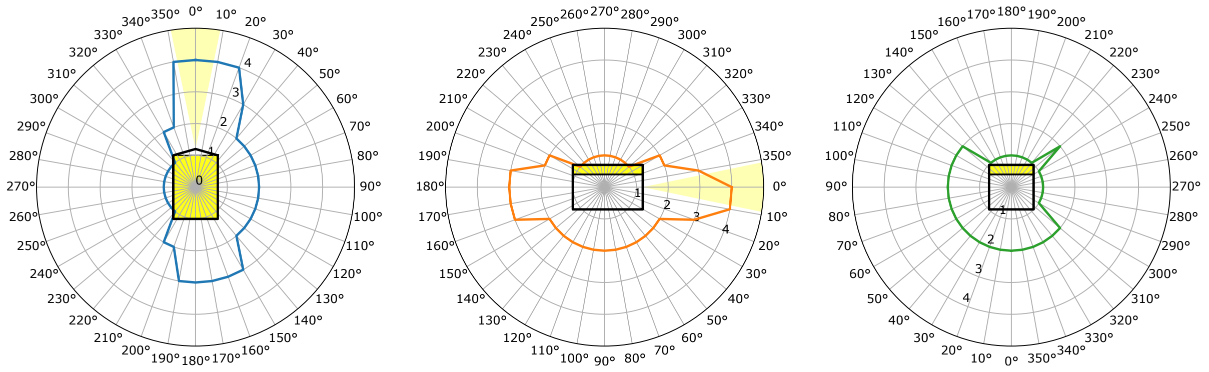


Figure 10.5. 3D ROV-DPCap plots of Minerva

shown in Fig. 10.7.



(a) XY

(b) XZ

(c) YZ

Figure 10.6. 2D ROV-DPcap plots of visor3

Figure 10.6 (a) shows horizontal plane motion capability. This plot evidences the asymmetries of surge and sway motions due to unidirectional thrusters. There is more capability for (surge) forward motion, with ROV-DPCap number around 4, than for backward motion, with ROV-DPCap number around 3. Conversely, there is more capability for starboard-side motion, with ROV-DPCap number around 2, than for port-side motion, with ROV-DPCap number around 1. As in Minerva, it is also apparent that there is more capability for overall surge motion

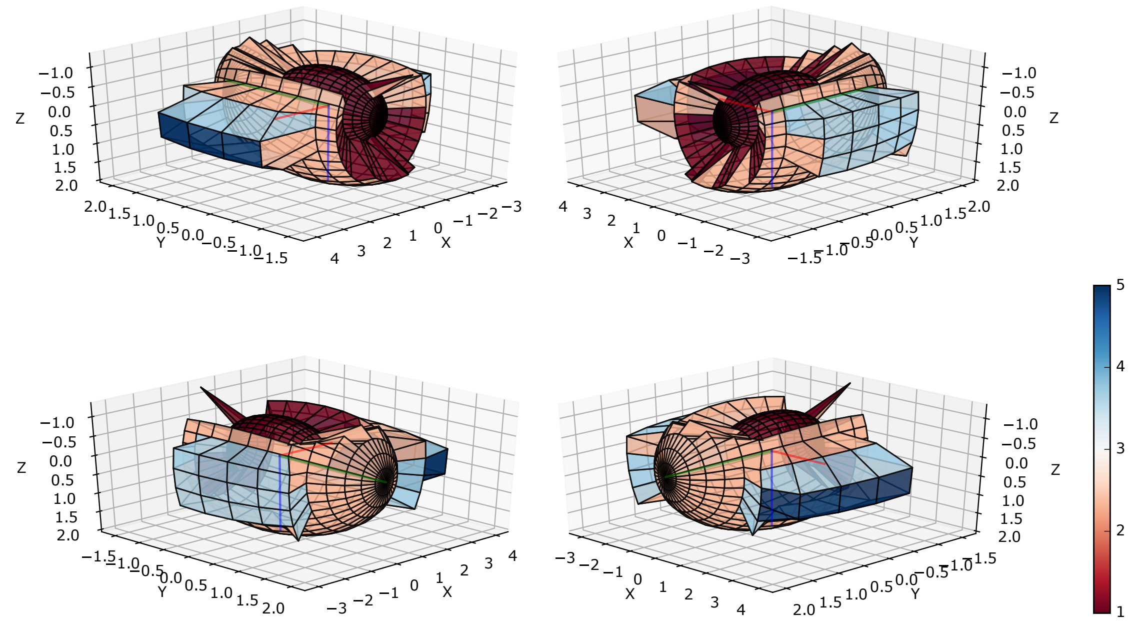


Figure 10.7. 3D ROV-DPCap plots of VISOR3

than for sway and heave motion, as it becomes apparent in xz- and yz-plots of Figs. 10.6 (b) and (c). Comparing the plots with those of MINERVA, this ROV has large unbalances of ROV-DPCap number. This means that even though it could withstand surge direction current in the range of 0.8 m/s in a best-case scenario, it could lose motion capability at small currents in any other direction.

10.4.3 L2 ANALYSIS OF TWO THRUSTER CONFIGURATIONS DURING ROV DESIGN

A ROV design-related analysis is addressed here through a minimalistic approach to L2 ROV-DPCap, *i.e.*, to elaborate 2D ROV-DPCap plots to compare two possible thruster configurations using simplified models for drag and thrust. In this case, the drag model in (2.12) has coefficients $X_{|u|u} = 206$, $Y_{|v|v} = 347$, and $Z_{|w|w} = 377$ Ns^{2}m^{-2} and he remaining coefficients are assumed zero. The thrusters are rated to have 166.6 N of bollard pull in the forward direction and 98 kg-f in the backward direction. For the sake of simplicity, it is assumed that thrust is not reduced due to advance ratio, *i.e.*, thrust coefficient is constant.

The two possible thruster configurations are as following: 1) 3 thrusters for motion in the

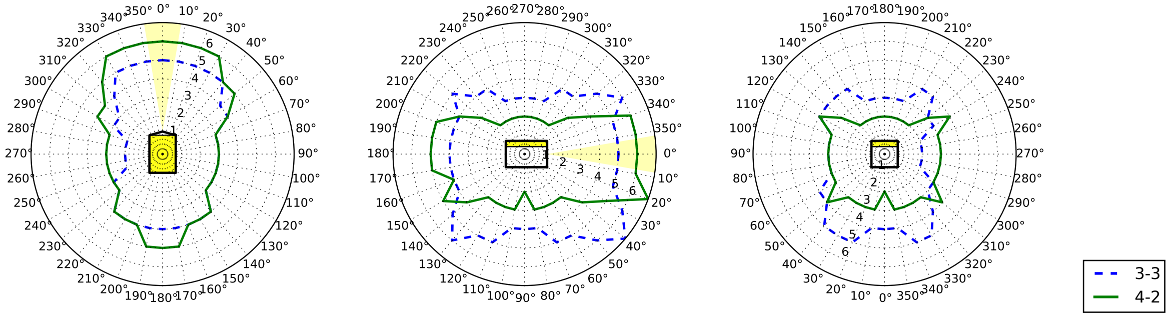
xy-plane and 3 thrusters for vertical motion, given by the configuration matrix in (2.16)

$$\mathrm{T} = \begin{bmatrix} 0.940 & 0.940 & 0 & 0 & 0 & 0.174 \\ 0.342 & -0.342 & 1 & -0.174 & 0.174 & 0 \\ 0 & 0 & 0 & 0.985 & 0.985 & 0.985 \\ 0 & 0 & 0 & 0.241 & -0.241 & 0 \\ 0 & 0 & 0 & -0.128 & -0.128 & 0.379 \\ -0.361 & 0.361 & 0.39 & -0.023 & 0.023 & 0 \end{bmatrix},$$

and 2) 4 thrusters for motion in the xy-plane and 2 thrusters for vertical motion, given configuration matrix (2.16),

$$\mathrm{T} = \begin{bmatrix} 0.819 & 0.819 & 0.819 & 0.819 & 0 & 0 \\ 0.574 & -0.574 & -0.574 & 0.574 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -0.24 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.42 \\ 0.007 & 0.007 & -0.007 & -0.007 & 0 & 0 \end{bmatrix}.$$

The two resulting 2D ROV-DPCap plots are summarised in Fig. 10.8. There, it is apparent how 3-3 configuration gives a more balanced but asymmetrical capability plot and 4-2 configuration gives more capability for motion in the xy-plane compared to vertical motion.



(a) XY (b) XZ (c) YZ

Figure 10.8. 2D DPCap plots of the ROV design

10.4.4 L2S ANALYSIS WHERE CABLE DRAG ACCUMULATION IS CONSIDERED

The effects of drag due to the cable is addressed here by using L2s ROV-DPCap analysis. To do so, the current profile of a location in the Colombian Caribbean is considered; more specifically, the profile at coordinates 12 N-73 W, from May 14th, 2014 (Fig. 10.9). This profile represents calm water conditions because current is always below 0.2 m/s.

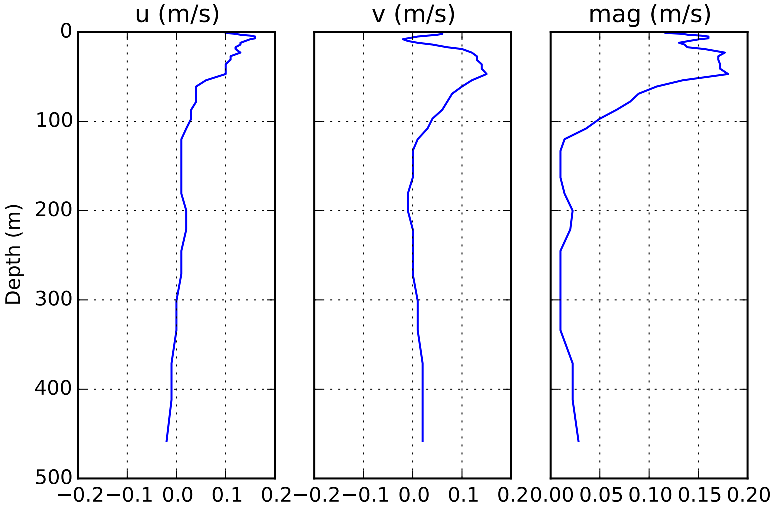


Figure 10.9. Current profile at 12 N-73 W.

This profile is obtained from the GLobal Ocean ReanalYses and Simulations (GLORYS2V3) of the EU-Copernicus's Marine Environment Monitoring Service (http://marine.copernicus.eu/) [45]. The analysis is done for 4-2 configuration, where it is more likely to lose vertical motion capability. Figure 10.10 shows a comparison between the xz-plane capability plots when the ROV is unaffected by the cable (datum) and when the ROV is submerged at 300 m and the cable accumulates drag. It is apparent that the ROV is able to maintain DP capability at

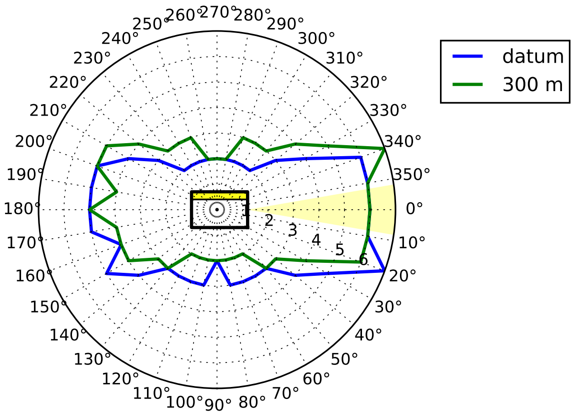


Figure 10.10. XZ DPCap plot at 300 m

the specified depth; likewise, one can also see the effect of the vertical component of the cable load where downward capability is reduced and upward capability increased.

10.5 CONCLUDING REMARKS

This thesis centred its objectives on developing a framework for assessing motion feasibility in ROVs; in accordance, this chapter stated that a convenient route for doing so is through the concept of dynamic positioning capability (DPCap). In this last chapter, the concept of ROV-DPCap was developed alongside the DNVGL-ST-0111 standard, which gives a framework for assessing DPCap on surface vessels. Consequently, developing a DPCap framework for ROVs required stating a definition of ROV-DPCap number, the types of analyses that can be pursued, and two- and three-dimensional graphical representations.

This work proposed a definition of ROV-DPCap number that is based on current. In other words, an ROV-DPCap number is indicative of the current magnitude the ROV can withstand. Therefore, DPCap number becomes a way to quantify motion feasibility; this means that it is possible to spot a situation where the operation is performed in a fragile condition, where the ROV is perhaps near to lose its capability of keeping position or course.

This chapter specifies how to perform ROV-DPCap analyses, *i.e.*, computing the ROV-DPCap number at certain foreknown conditions. Similarly as in the DNVGL-ST-0111 standard, the different types of analyses include all combinations of pseudo-static and dynamic motion conditions, which can be evaluated at standard or site-dependent operation conditions. Standard conditions represent those were current-induced load is the only external stimulus; in contrast, other external stimuli derive in site-dependent conditions. For instance, cable effects, waves, or any other depth-dependent effects are regarded as site-dependent conditions. Correspondingly, the considerations for each type of analysis were explained in terms of the ROV's dynamics model, and a general algorithm is presented to compute the corresponding ROV-DPCap number.

It was acknowledged that ROV-DPCap analysis is essentially three-dimensional; this is stated in contrast to the fact that two-dimensional analyses suffice for surface vessels. This is important to note, because the distribution of different orientations (directions) in three dimensions is more intricate than that in two dimensions, when even distribution is a concern. Nevertheless, representation is two dimensions are valued as well because they are easier to interpret.

In order to illustrate how ROV-DPCap analyses work, some case studies were computed. These computations are intended to illustrate the potentiality of this approach to ROV motion feasibility analysis. They are regarded, in consequence, as a passageway to further work.

With everything considered, the prospects of motion feasibility are wide. The spherical geometry of the problem could be further exploited. The impact of the cable on motion feasibility requires detailed attention. A reasonable next step would be to incorporate motion feasibility tools into ROV operations planning. And, so far, the type of studies are intended to be performed in an offline fashion; this means that there is plenty of space to extend this work to online analyses.

Part IV

Epilogue

Conclusions and further work

CONCLUSIONS

This thesis proposed a framework for motion feasibility studies on ROVs based on time-domain numerical solutions to models that span rigid-body dynamics, control algorithms, and elastic rod dynamics. Hence, the thesis benefited from known rigid-body dynamics and control theory and required emphasizing on modelling hydrodynamics and cable loads. Furthermore, the thesis incorporated as well tools such as sphere geometry, design of computer experiments, and object-oriented programming, which are all in all a fitting complement. The problem of motion feasibility was addressed first by considering time-domain manoeuvring simulations. Subsequently, with everything taken into account, a sound framework for analysing motion feasibility was developed from the concept of ROV Dynamic Positioning Capability (ROV-DPCap). To the author's knowledge, the concept of ROV motion feasibility based on ROV-DPCap, as well as a framework that integrates different motion-relevant components, has not been treated previously; consequently, this was introduced as part of the author's research.

Modelling ROV hydrodynamics played a central role in this thesis. Here, a methodology for obtaining a meta-model for the hydrodynamic forces and moments that act on an ROV and is useful in time-domain simulations was proposed. It is assumed that the source of data are viscous-flow computations performed in available CFD code. More concretely, the complete process was carried out for VISOR3, where the viscous-flow computations were performed in MARIN's viscous-flow solver ReFRESCO. This work emphasized the use of the language of design of computer experiments and quantifying numerical uncertainty due to different sources. Studying manoeuvring is commonplace for ships, but for ROVs is a ground that is worth exploring through more research. In particular, this thesis stated that coming up with simplified manoeuvring meta-models from viscous-flow computations requires addressing design of computer experiments for spherical domains, the geometry of basis functions, and uncertainty due to the meta-model itself as a major source of inaccuracy.

Including the effects of a cable into ROV motion computations was a purpose of this thesis. This topic proved to be unbounded given the numerous possibilities to approaching the model. Acknowledging this, this thesis posited that it was more convenient, given this work's limitations, to establish a starting point that allows one to build-up further knowledge. Such point of departure was the case of the classical catenary, which gives the basis for solving problems were, *e.g.*, drag accumulation is computed. The so-called drag-catenary model was approached by using different numerical methods, namely shooting-algorithm, finite-differences, finite-elements method. Elastic effects were approached and analysed by performing computations on the WHOI cable model, both in steady-state and dynamic conditions.

The manoeuvring of ROVs was studied by performing computations on VISOR3's model. These computations were carried through on a tailor-made, object-oriented framework for ROV time-domain simulation. The manoeuvring response of the ROV at different scenarios was done in three steps: open-loop without thrust allocation, open-loop with thrust allocation, and multi-loop PID control. The different scenarios include both calm-water and oblique-current manoeuvres, by simulating, *e.g.*, the response of all controllable degrees of freedom (surge, sway, heave, and yaw) or circular motion. These simulations proved to be meaningful and effective for gaining insight into the ROV's manoeuvring attributes.

The concept of ROV-DPCap was put forward as a means for analysing and quantifying motion feasibility. Acknowledging the fact that DPCap is commonplace on surface vessels, this thesis focused on tailoring the method for ROV particularities. This comprised the definition of ROV-DPCap number based on three-dimensional relative current as the main load-inducing effect; specifying the types of analyses that can be pursued considering quasi-static and dynamic conditions, as well as standard and site-dependent load conditions; and two- and three-dimensional graphical representations. ROV-DPCap constitutes a promising topic for further research on ROVs. To the author's knowledge, DPCap studies for ROVs are a contribution that resulted within the author's research; it constitutes an innovative approach to assessing motion feasibility for ROVs.

FURTHER WORK

An inevitable pathway to further work is to perform comprehensive experimentation. Because this work was integrative, experimentation should range from testing individual models, such as hydrodynamics and cable, to different assembly of different components.

The topic reach of this thesis may become fractal. Most of the topics discussed here are susceptible to be further researched in detail, namely rigid body dynamics; hydrodynamics, CFD, and manoeuvring; cable and, in general, elastic components; (sphere) geometry and design of computer experiments; and (undoubtedly) ROV-DPCap.

From all topics, cable modelling is particularly fractal and requires detailed attention. From all possibilities, one that leads to simulations that make a compromise between computational effort and accuracy seems outstandingly appealing. Moreover, another possibility of particular practical interest is studying the possibility of seeing the cable as a controllable object.

The implications of designing more elaborate feedback control when there is better knowledge on the ROV model (like hydrodynamics and cable) is to be addressed: it may improve features such as manoeuvrability, energy consumption, and the scope of underwater operations.

A natural next step to ROV-DPCap analysis is to incorporate such analyses into ROV operation planning; and, furthermore, to extend it to become an online advisory tool.

References

[1] C.M. Ablow and S. Schechter. Numerical simulation of undersea cable dynamics. *Ocean Engineering*, 10(6):443–457, 1983.

[2] D.J. Amon, A.F. Ziegler, T.G. Dahlgren, A.G. Glover, A. Goineau, A.J. Gooday, H. Wiklund, and C.R. Smith. Insights into the abundance and diversity of abyssal megafauna in a polymetallic-nodule region in the eastern Clarion-Clipperton Zone. *Scientific Reports*, 6, 2016.

[3] W. Antheunisse, D. Chalkias, and R. Huijsmans. Offset requirements in quasi static DP capability calculations. In *Proceedings of the International Offshore and Polar Engineering Conference*, volume 2016-January, pages 1602–1608, 2016.

[4] S.S. Antman. *Mechanics of Solids, Vol. II*, chapter The Theory of Rods, pages 641–704. Springer, 1984.

[5] M. Araki, H. Sadat-Hosseini, Y. Sanada, K. Tanimoto, N. Umeda, and F. Stern. Estimating maneuvering coefficients using system identification methods with experimental, system-based, and CFD free-running trial data. *Ocean Engineering*, 51:63–84, 2012.

[6] M.S.M. Aras, S.S. Abdullah, M.Z.A. Rashid, A.A. Rahman, and M.A.A. Aziz. Development and modeling of unmanned underwater remotely operated vehicle using system identification for depth control. *Journal of Theoretical and Applied Information Technology*, 56(1):136–145, 2013.

[7] L.M. Aristizábal, S. Rúa, C.E. Gaviria, S.P. Osorio, C.A. Zuluaga, N.L. Posada, and R.E. Vásquez. Design of an open source-based control platform for an underwater remotely operated vehicle [Diseño de una plataforma de control basada en fuente abierta para un vehículo subacuático operado remotamente]. *DYNA (Colombia)*, 83(195):198–205, 2016.

[8] L.M. Aristizábal and C.A. Zuluaga. Distance measure with computer vision and neural

networks for underwater applications. In *IEEE Colombian Conference on Robotics and Automation, CCRA 2016*, 2017.

[9] J.P.J Avila and J.C. Adamowski. Experimental evaluation of the hydrodynamic coefficients of a ROV through morison's equation. *Ocean Engineering*, 38(17-18):2162 – 2170, 2011.

[10] J.P.J Avila, D.C. Donha, and J.C. Adamowski. Experimental model identification of open-frame underwater vehicles. *Ocean Engineering*, 60:81 – 94, 2013.

[11] J.G. Balchen, N.A. Jenssen, E. Mathisen, and S. Saelid. Dynamic positioning system based on Kalman filtering and optimal control. *Modeling, Identification and Control*, 1(3):135–163, 1980.

[12] M. Bateman and P. Wallace. DP design philosophy and planning for assurance success. In *Proceedings of the Annual Offshore Technology Conference*, volume 5, pages 3304–3314, 2015.

[13] A. Bliek. *Dynamic analysis of single span cables.* PhD thesis, Dept. of Ocean Engineering, MIT, 1984.

[14] M. Bonci, M. Viviani, R. Broglia, and G. Dubbioso. Method for estimating parameters of practical ship manoeuvring models based on the combination of RANSE computations and system identification. *Applied Ocean Research*, 52:274–294, 2015.

[15] N. Bou-Rabee and J.E. Marsden. Hamilton-Pontryagin integrators on Lie groups part I: Introduction and structure-preserving properties. *Foundations of Computational Mathematics*, 9(2):197–219, 2009.

[16] F. Boyer, G. De Nayer, A. Leroyer, and M. Visonneau. Geometrically exact Kirchhoff beam theory: Application to cable dynamics. *Journal of Computational and Nonlinear Dynamics*, 6(4), 2011.

[17] J.S. Brauchart and P.J. Grabner. Distributing many points on spheres: Minimal energy and designs. *Journal of Complexity*, 31(3):293–326, 2015.

[18] C. Brun, D. Coache, Y. Giorgiutti, and F. Rezende. Influences on dynamic positioning system capability. In *Proceedings of the International Offshore and Polar Engineering Conference*, volume 2015-January, pages 627–634, 2015.

[19] B. Buckham, F.R. Driscoll, and M. Nahon. Development of a finite element cable model for use in low-tension dynamics simulation. *Journal of Applied Mechanics, Transactions ASME*, 71(4):476–485, 2004.

[20] M. Candeloro. *Tools and Methods for Autonomous Operations on Seabed and Water*

Column using Underwater Vehicles. PhD thesis, Norwegian University of Science and Technology (NTNU), 2016.

[21] M. Candeloro, F. Mosciaro, A.J. Sørensen, G. Ippoliti, and M. Ludvigsen. Sensor-based autonomous path-planner for sea-bottom exploration and mosaicking. *IFAC-PapersOnLine*, 28(16):31–36, 2015.

[22] M. Candeloro, E. Valle, M.R. Miyazaki, R. Skjetne, M. Ludvigsen, and A.J. Sørensen. HMD as a new tool for telepresence in underwater operations and closed-loop control of ROVs. In *OCEANS 2015 - MTS/IEEE Washington*, 2016.

[23] P.M. Carrica, A. Mofidi, K. Eloot, and G. Delefortrie. Direct simulation and experimental study of zigzag maneuver of KCS in shallow water. *Ocean Engineering*, 112:117–133, 2016.

[24] C. Chin and M. Lau. Modeling and testing of hydrodynamic damping model for a complex-shaped remotely-operated vehicle for control. *Journal of Marine Science and Application*, 11(2):150–163, 2012.

[25] C.S. Chin and S.H. Lum. Rapid modeling and control systems prototyping of a marine robotic vehicle with model uncertainties using xPC target system. *Ocean Engineering*, 38(17-18):2128 – 2141, 2011.

[26] R.D. Christ and R.L. Wernli. *The ROV Manual, a user guide for Remotely Operated Vehicles.* Butterworth-Heinemann, Oxford, 2014.

[27] J. Chung and G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *Journal of Applied Mechanics, Transactions ASME*, 60(2):371–375, 1993.

[28] R.D. Cook, D.S. Malkus, M.E. Plesha, and R.J. Witt. *Concepts and applications of Finite Element Analysis, 4th ed.* John Wiley & Sons, 2001.

[29] J.T. Copley, L. Marsh, A.G. Glover, V. H´uhnerbach, V.E. Nye, W.D.K. Reid, C.J. Sweeting, B.D. Wigham, and H. Wiklund. Ecology and biogeography of megafauna and macrofauna at the first known deep-sea hydrothermal vents on the ultraslow-spreading Southwest Indian Ridge. *Scientific Reports*, 6, 2016.

[30] J.C. Correa, N. Restrepo, J.A. Ramírez, E.A. Taborda, R.E. Vasquez, C.A. Zuluaga, J.M. Londo no, D.A. Flórez, and E. Palacio. Diseño de un vehículo subacuático operado remotamente. In *XVII Seminario Nacional de Ciencias y TecnologÃas Del Mar, SENALMAR*, 2017.

[31] J.C. Correa, R.E. Vásquez, J.A. Ramírez-Macías, E.A. Taborda, C.A. Zuluaga, N.L. Posada, and J.M. Londoño. Una arquitectura para el diseño conceptual de vehículos

para exploración subacuática. *Ingeniería Y Ciencia*, 2015.

[32] D. de A. Fernandes. *An output feedback motion control system for ROVs.* PhD thesis, Norwegian University of Science and Technology (NTNU), 2015.

[33] DNV-GL. Recommended practice DNV-RP-C205: Environmental conditions and environmental loads. Technical report, DNV-GL, October 2010.

[34] DNV-GL. Standard DNVGL-ST-0111: Assessment of station keeping capability of dynamic positioning vessels. Technical report, DNV-GL, July 2016.

[35] F.R. Driscoll, R.G. Lueck, and M. Nahon. Development and validation of a lumped-mass dynamics model of a deep-sea ROV system. *Applied Ocean Research*, 22(3):169–182, 2000.

[36] F. Dukan. *ROV Motion Control Systems.* PhD thesis, Norwegian University of Science and Technology (NTNU), 2014.

[37] L. Eca and M. Hoekstra. Verification and validation for marine applications of CFD. In *29th Symposium on Naval Hydrodynamics (ONR)*, Gothenburg, Sweden, August 2012.

[38] L. Eça and M. Hoekstra. A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies. *Journal of Computational Physics*, 262:104–130, 2014.

[39] L. Eça, C.M. Klaij, G. Vaz, M. Hoekstra, and F.S. Pereira. On code verification of RANS solvers. *Journal of Computational Physics*, 310:418–439, 2016.

[40] L. Eça, G. Vaz, and M. Hoekstra. Code verification, solution verification and validation in RANS solvers. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, volume 6, pages 597–605, 2010.

[41] D.M. Edessa, L. Kleinsorge, and R. Bronsart. Assuring quality ship hull form representation for downstream applications. In *Proceedings of MARTECH 2014: 2nd International Conference on Maritime Technology and Engineering*, volume 1, pages 317–324, 2015.

[42] EIU. The blue economy: Growth, opportunity and a sustainable ocean economy. Technical report, The Economist Intelligence Unit, 2015.

[43] O. Faltinsen. *Sea loads on ships and offshore structures.* Cambridge University Press, 1990.

[44] K.T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments.* Chapman & Hall/CRC, 2006.

[45] N. Ferry, L. Parent, S. Masina, A. Storto, H. Zuo, and M. Balmaseda. *Product User Manual For Global Ocean Reanalysis Products GLOBAL-REANALYSIS-PHYS-001-009, -011, and -017.* COPERNICUS, 2.9 edition, January 2016.

[46] R. Folb and J.J. Nelligan. Hydrodynamic loading on armored towcables. Technical report, David W. Taylor Naval Ship Research and Development Center, 1983.

[47] J. Foley, D. Jennings, R. Fonda, J. Jacobson, and M. Miele. Improved capability, reliability, and productivity for underwater geophysical mapping of unexploded ordnance. In *MTS/IEEE OCEANS 2015*, 2015.

[48] T.I. Fossen. *Handbook of marine craft hydrodynamics and motion control.* John Wiley & Sons, West Sussex, United Kingdom, 2011.

[49] D.L. Garrett. Dynamic analysis of slender rods. *Journal of Energy Resources Technology, Transactions of the ASME*, 104(4):302–306, 1982.

[50] S.S. Garud, I.A. Karimi, and M. Kraft. Design of computer experiments: A review. *Computers and Chemical Engineering*, 106:71–95, 2017.

[51] Y. Giorgiutti, F.C. Rezende, J. Boulland, and R. Araujo. The impact of multibody operations on DP capability. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, 2015.

[52] J.I. Gobat and M.A. Grosenbaugh. WHOI cable: Time domain numerical modeling of moored and towed oceanographic systems. In *Proceedings of the 1998 Oceans Conference*, volume 3, pages 1681–1685, 1998. cited By 2.

[53] J.I. Gobat and M.A. Grosenbaugh. Application of the generalized-α method to the time integration of the cable dynamics equations. *Computer Methods in Applied Mechanics and Engineering*, 190(37-38):4817–4829, 2001.

[54] J.I. Gobat and M.A. Grosenbaugh. Time-domain numerical simulation of ocean cable structures. *Ocean Engineering*, 33(10):1373–1400, 2006.

[55] S.C.P. Gomes, E.B. Zanela, and A.E.L. Pereira. Automatic generation of dynamic models of cables. *Ocean Engineering*, 121:559–571, 2016.

[56] A. Gómez, L.M. Aristizábal, C.A. Zuluaga, J.C. Correa, and R.E. Vásquez. Development and implementation of a high-level control system for the underwater remotely operated vehicle Visor3. *IFAC-PapersOnLine*, 50(1):1151–1156, 2017.

[57] L.B. Gutiérrez, C.A. Zuluaga, J.A. Ramírez, R.E. Vásquez, D.A. Flórez, E.A. Taborda, and R.A. Valencia. Development of an underwater remotely operated vehicle (ROV) for surveillance and inspection of port facilities. In *Proceedings of the ASME International*

Mechanical Engineering Congress and Exposition (IMECE), volume 11, pages 631–640, 2010.

[58] B.S. Halpern, M. Frazier, J. Potapenko, K.S. Casey, K. Koenig, C. Longo, J.S. Lowndes, R.C. Rockwood, E.R. Selig, K.A. Selkoe, and S. Walbridge. Spatial and temporal changes in cumulative human impacts on the world's ocean. *Nature Communications*, 6, 2015.

[59] B.S. Halpern, C. Longo, D. Hardy, K.L. McLeod, J.F. Samhouri, S.K. Katona, K. Kleisner, S.E. Lester, J. Oleary, M. Ranelletti, A.A. Rosenberg, C. Scarborough, E.R. Selig, B.D. Best, D.R. Brumbaugh, F.S. Chapin, L.B. Crowder, K.L. Daly, S.C. Doney, C. Elfes, M.J. Fogarty, S.D. Gaines, K.I. Jacobsen, L.B. Karrer, H.M. Leslie, E. Neeley, D. Pauly, S. Polasky, B. Ris, K. St Martin, G.S. Stone, U. Rashid Sumaila, and D. Zeller. An index to assess the health and benefits of the global ocean. *Nature*, 488(7413):615–620, 2012.

[60] R.H. Hardin and N.J.A. Sloane. Codes (spherical) and designs (experimental). *Proceedings of Symposia in Applied Mathematics*, 1995.

[61] J. Hawkes, G. Vaz, S. Turnock, S. Cox, and A. Philips. Software performance analysis of massively-parallel hydrodynamics simulations. In *11th International Conference on Hydrodynamics (ICHD)*, Singapore, October 2014.

[62] C.T. Howell. *Investigations of the Dynamics of Low-Tension Cables.* PhD thesis, Woods Hole Oceanographic Institution, MIT, 1992.

[63] IMCA. IMCA M 140 specification for DP capability plots. Technical report, The International Marine Contractors Association, 2000.

[64] H.M. Irvine. *Cable structures.* The MIT Press, 1981.

[65] P. Jagadeesh, K. Murali, and V.G. Idichandy. Experimental investigation of hydrodynamic force coefficients over AUV hull form. *Ocean Engineering*, 36(1):113–118, 2009.

[66] V. Johansen, S. Ersdal, A.J. Sørensen, and B. Leira. Modelling of inextensible cable dynamics with experiments. *International Journal of Non-Linear Mechanics*, 41(4):543–555, 2006.

[67] J.W. Kamman and R.L. Huston. Modelling of submerged cable dynamics. *Computers and Structures*, 20(1-3):623–629, 1985.

[68] S. Kerkeni, I. Metrikin, and P. Jochmann. Capability plots of dynamic positioning in ice. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, volume 6, 2013.

[69] M. Kerkvliet, G. Vaz, N. Carette, and M. Gunsing. Analysis of U-Type Anti-Roll Tanks using RANS. Sensitivity and Validation. In *OMAE2014*, San-Francisco, USA, June 2014.

[70] H. Kim, H. Akimoto, and H. Islam. Estimation of the hydrodynamic derivatives by RaNS simulation of planar motion mechanism test. *Ocean Engineering*, 108:129–139, 2015.

[71] C. M. Klaij and C. Vuik. Simple-type preconditioners for cell-centered, colocated finite volume discretization of incompressible Reynolds-averaged Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 71(7):830–849, 2013.

[72] M. Kobilarov, K. Crane, and M. Desbrun. Lie group integrators for animation and control of vehicles. *ACM Transactions on Graphics*, 28(2), 2009.

[73] A. H. Koop, C. M. Klaij, and G. Vaz. *MARINE 2011, IV International Conference on Computational Methods in Marine Engineering: Selected Papers*, chapter Viscous-Flow Calculations for Model and Full-Scale Current Loads on Typical Offshore Structures, pages 3–29. Springer Netherlands, Dordrecht, 2013. Editor: Eça, Luís and Oñate, Eugenio and García-Espinosa, Julio and Kvamsdal, Trond and Bergan, Pål.

[74] G. Kuiper. *The Wageningen Propeller Series*. MARIN, 1992.

[75] S. Levy and D.M. Steinberg. Computer experiments: A review. *AStA Advances in Statistical Analysis*, 94(4):311–324, 2010.

[76] A.E.H. Love. *A treatise on the mathematical theory of elasticity*. Cambridge University Press, 1892.

[77] M. Ludvigsen. *An ROV toolbox for optical and acoustical seabed investigations*. PhD thesis, Norwegian University of Science and Technology (NTNU), 2010.

[78] M. Ludvigsen and A.J. Sørensen. Towards integrated autonomous underwater operations for ocean mapping and monitoring. *Annual Reviews in Control*, 42:145–157, 2016.

[79] A. Luebcke, P. Augener, and A. Falkenhorst. A fast method for the evaluation of the dynamic positioning capability in time domain in the early design stage. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, volume 1, 2015.

[80] W. Luo, L. Moreira, and C. Guedes Soares. Manoeuvring simulation of catamaran by using implicit models based on support vector machines. *Ocean Engineering*, 82:150–159, 2014.

[81] A.B. Mahfouz. Predicting the capability-polar-plots for dynamic positioning systems for offshore platforms using artificial neural networks. *Ocean Engineering*, 34(8-9):1151–1163, 2007.

[82] A.B. Mahfouz and H.W. El-Tahan. On the use of the capability polar plots program for dynamic positioning systems for marine vessels. *Ocean Engineering*, 33(8-9):1070–1089, 2006.

[83] N. Newman. *Marine Hydrodynamics*. MIT Press, 1977.

[84] N. Newmark. A method of computation for structural dynamics. *Journal of Engineering Mechanics, ASCE*, 85 (EM3):67–94, 1959.

[85] D.T. Nguyen, L. Pivano, and Ø. Smogeli. Estimation of dynamic positioning performance by time-domain simulations—a step toward safer operations. In *Proceedings of the International Conference on Maritime and Port Technology and Development, MTEC 2014*, pages 109–116, 2015.

[86] I. Nilssen, T. Ødegård, A.J. Sørensen, G. Johnsen, M.A. Moline, and J. Berge. Integrated environmental mapping and monitoring, a methodological approach to optimise knowledge gathering and sampling strategy. *Marine Pollution Bulletin*, 96(1-2):374–383, 2015.

[87] R.P. Nordgren. On computation of the motion of elastic rods. *Journal of Applied Mechanics, Transactions ASME*, 41(3):777–780, 1974.

[88] N. Nordkvist and A.K. Sanyal. A Lie group variational integrator for rigid body motion in SE(3) with applications to underwater vehicle dynamics. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5414–5419, 2010.

[89] D. Obreja, R. Nabergoj, L. Crudu, and S. Păcuraru-Popoiu. Identification of hydrodynamic coefficients for manoeuvring simulation model of a fishing vessel. *Ocean Engineering*, 37(8-9):678–687, 2010.

[90] Ø. Ødegård, A.J. Sørensen, R.E. Hansen, and M. Ludvigsen. A new method for underwater archaeological surveying using sensors and unmanned platforms. *IFAC-PapersOnLine*, 49(23):486–493, 2016.

[91] OECD. *The Ocean Economy in 2030*. OECD Publishing, 2016.

[92] S.P. Osorio, L.M. Aristizábal, and C.A. Zuluaga. Development of a command interface based on handheld devices for remotely operated vehicles. In *IEEE Colombian Conference on Robotics and Automation, CCRA 2016*, 2017.

[93] A.E.L. Pereira, S.C.P. Gomes, and Á.L. De Bortoli. A new formalism for the dynamic

modelling of cables. *Mathematical and Computer Modelling of Dynamical Systems*, 19(3):263–276, 2013.

[94] F. Pereira, L. Eça, and G. Vaz. On the Order of Grid Convergence of the Hybrid Convection Scheme for RANS Codes. In *CMNI2013*, pages 1–21, Barcelona, Spain, 2013.

[95] F.S. Pereira, L. Eça, and G. Vaz. Verification and validation exercises for the flow around the KVLCC2 tanker at model and full-scale Reynolds numbers. *Ocean Engineering*, 129:133–148, 2017.

[96] L. Pronzato and W.G. Müller. Design of computer experiments: Space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012.

[97] W.-C. Quan, Z.-Y. Zhang, A.-Q. Zhang, Q.-F. Zhang, and Y. Tian. A geometrically exact formulation for three-dimensional numerical simulation of the umbilical cable in a deep-sea ROV system. *China Ocean Engineering*, 29(2):223–240, 2015.

[98] J.A. Ramírez-Macías, P. Brongers, S. Rúa, and R.E. Vásquez. Hydrodynamic modelling for the remotely operated vehicle Visor3 using CFD. *IFAC-PapersOnLine*, 49(23):187–192, 2016.

[99] J.A. Ramírez-Macías, P. Brongers, and R.E. Vásquez. On the use of time-domain simulation in the design of remotely operated vehicles. *Ship Science & Technology*, 11(21):41–49, July 2017.

[100] J.A. Ramírez-Macías, L.B. Gutiérrez, R.E. Vásquez, and D.A. Flórez. Mechanical/naval design of an underwater remotely operated vehicle (ROV) for surveillance and inspection of port facilities. In *Proceedings of the ASME International Mechanical Engineering Congress and Exposition,*, volume 16, pages 351–360, 2008.

[101] J.A. Ramírez-Macías, R.E. Vásquez, A.J. Sørensen, and S. Sævik. A methodology for DP capability studies on remotely operated vehicles. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, volume 7A-2017, 2017.

[102] G.T. Reilly and M. Hensley. Dynamic positioning capability and enhanced reliability. In *Proceedings of the Annual OTC*, volume 2, pages 1406–1417, 2011.

[103] E. Revestido-Herrero and F.J. Velasco-González. Two-step identification of non-linear manoeuvring models of marine vessels. *Ocean Engineering*, 53:72–82, 2012.

[104] D. Rijpkema and G. Vaz. Viscous flow computations on propulsors: verification, validation and scale effects. In *RINA 2011*, 2011.

[105] G. F. Rosetti, G. Vaz, and A.L.C. Fujarra. URANS Calculations for Smooth Circular Cylinder Flow in a Wide Range of Reynolds Numbers: Solution Verification and Validation. *Journal of Fluids Engineering, ASME*, page 549, July 2012.

[106] S. Rúa and R.E. Vásquez. Development of a low-level control system for the ROV Visor3. *International Journal of Navigation and Observation*, 2016, 2016.

[107] M.B. Rubin. *Cosserat theories: shells, rods and points.* Springer, 2000.

[108] P. Rundtop and K. Frank. Experimental evaluation of hydroacoustic instruments for ROV navigation along aquaculture net pens. *Aquacultural Engineering*, 74:143–156, 2016.

[109] E.B. Saff and A.B.J. Kuijlaars. Distributing many points on a sphere. *Mathematical Intelligencer*, 19(1):5–11, 1997.

[110] ScyPy. ScyPy Reference Guide — scipy.integrate.ode. Online, 2016.

[111] ScyPy. ScyPy Reference Guide — scipy.optimize.fsolve. Online, 2016.

[112] J.-J. Serraris. Time domain analysis for DP simulations. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, volume 1, pages 595–605, 2009.

[113] M. Servin and C. Lacoursière. Rigid body cable for virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):783–796, 2008.

[114] J.C. Simo and L. Vu-Quoc. On the dynamics in space of rods undergoing large motions - a geometrically exact approach. *Computer Methods in Applied Mechanics and Engineering*, 66(2):125–161, 1988.

[115] N. Sloane, R. Hardin, W. Smith, et al. Tables of spherical codes: Coverings. Online: http://neilsloane.com/coverings/, April 2018.

[116] N. Sloane, R. Hardin, W. Smith, et al. Tables of spherical codes: Minimal energy. Online: http://neilsloane.com/electrons/, April 2018.

[117] N. Sloane, R. Hardin, W. Smith, et al. Tables of spherical codes: Packings. Online: http://neilsloane.com/packings/, April 2018.

[118] N. Sloane, R. Hardin, W. Smith, et al. Tables of spherical codes: t-designs. Online: http://neilsloane.com/sphdesigns/, April 2018.

[119] Ø. Smogeli, D.T. Nguyen, B. Børhaug, and L. Pivano. The next level DP capability analysis. In *Maritime Technology Society Dynamic Positioning Conference*, October 2013.

[120] A. Sørensen. *Marine Control Systems: Propulsion and Motion Control Systems of Ships and Ocean Structures*. Department of Marine Technology, NTNU, 2013.

[121] A.J. Sørensen. A survey of dynamic positioning control systems. *Annual Reviews in Control*, 35(1):123–136, 2011.

[122] A.J. Sørensen, F. Dukan, M. Ludvigsen, D. de Almeida Fernandes, and M. Candeloro. *Development of dynamic positioning and tracking system for the ROV Minerva*, chapter 6, pages 113–128. Institution of Engineering and Technology, 2012. Editor: Roberts, G.N. and Sutton, R.

[123] S. Sutulo and C. Guedes Soares. An algorithm for offline identification of ship manoeuvring mathematical models from free-running tests. *Ocean Engineering*, 79:10–25, 2014.

[124] S. Tang, T. Ura, T. Nakatani, B. Thornton, and T. Jiang. Estimation of the hydrodynamic coefficients of the complex-shaped autonomous underwater vehicle TUNA-SAND. *Journal of Marine Science and Technology*, 14(3):373–386, 2009.

[125] M. Tournier, M. Nesme, B. Gilles, and F. Faure. Stable constrained dynamics. *ACM Transactions on Graphics*, 34(4), 2015.

[126] S. Toxopeus. *Practical application of viscous-flow calculations for the simulation of manoeuvring ships*. PhD thesis, TU Delft, 2011.

[127] S. Toxopeus and G. Vaz. Calculation of current or manoeuvring forces using a viscous-flow solver. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering*, volume 5, pages 717–728, 2009.

[128] S.L. Toxopeus, C.D. Simonsen, E. Guilmineau, M. Visonneau, T. Xing, and F. Stern. Investigation of water depth and basin wall effects on KVLCC2 in manoeuvring motion using viscous-flow calculations. *Journal of Marine Science and Technology*, 18(4):471–496, 2013.

[129] M.S. Triantafyllou and F.S. Hover. *Maneuvering and Control of Marine Vehicles*. Massachusetts Institute of Technology, 2003.

[130] J. Unpingco. *Python for Probability, Statistics, and Machine Learning*. Springer, 2016.

[131] R.A. Valencia, J.A. Ramírez, L.B. Gutiérrez, and M.J. García. Modeling and simulation of an underwater remotely operated vehicle (ROV) for surveillance and inspection of port facilities using CFD tools. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, volume 4, pages 329–338, 2008.

[132] R. Van 't Veer and M. Gachet. Dynamic positioning — early design, capability and offsets, a novel approach. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, volume 3, pages 755–764, 2011.

[133] R.E. Vásquez, C.A. Zuluaga, E.A. Taborda, J.A. Ramírez-Macías, P. Zapata, and J.C. Correa. Development and use of unmanned vehicles for coastal/marine ecosystems characterization: Seaflower scientific expedition 2016. In *XVII Seminario Nacional de Ciencias y TecnologÃas Del Mar, SENALMAR*, 2017.

[134] G. Vaz, F. Jaouen, and M. Hoekstra. Free-surface viscous flow computations. validation of URANS code FreSCo. In *OMAE2009*, Hawaii, USA, May 31- June 5 2009.

[135] G. Vaz, S. Toxopeus, and Samuel Holmes. Calculation of manoeuvring forces on submarines using two viscous-flow solvers. In *ASME 2010 29th International Conference on Ocean, Offshore and Arctic Engineering OMAE2010*, pages 621–633, 2010.

[136] F.J. Velasco, E. Revestido-Herrero, F.J. Lastra-Santos, J.M. Riola-Rodriguez, J.J. Diaz-Hernandez, and L.M. Vega-Antolin. Measurements of hydrodynamic parameters and control of an underwater torpedo-shaped vehicle. *IFAC-PapersOnLine*, 48(2):167 – 172, 2015.

[137] M. Visbeck, U. Kronfeld-Goharani, B. Neumann, W. Rickels, J. Schmidt, E. van Doorn, N. Matz-Lück, K. Ott, and M.F. Quaas. Securing blue wealth: The need for a special sustainable development goal for the ocean and coasts. *Marine Policy*, 48:184–191, 2014.

[138] R. von Schwerin. *Multibody System Simulation: numerical methods, algorithms, and software.* Springer, 1999.

[139] C. Wang, F. Zhang, and D. Schaefer. Dynamic modeling of an autonomous underwater vehicle. *Journal of Marine Science and Technology*, 20(2):199–212, 2014.

[140] D.L. Webb. DP and operability capabilities of the dynamically positioned drillship Ocean Clipper. In *Proceedings of the IADC/SPE Asia Pacific Drilling Technology Conference, APDT*, pages 373–408, 1998.

[141] J. Windt. Adaptive mesh refinement in viscous flow solvers: Refinement in the near-wall region, implementation and verification. In *NUTTS2013*, Mulheim, Germany., 2013.

[142] P. Wriggers. *Nonlinear Finite Element Methods.* Springer, 2008.

[143] S. Xu, L. Wang, and X. Wang. Local optimization of thruster configuration based on a synthesized positioning capability criterion. *International Journal of Naval Architecture and Ocean Engineering*, 7(6):1044–1055, 2015.

[144] S. Xu, X. Wang, L. Wang, S. Meng, and B. Li. A thrust sensitivity analysis based on a synthesized positioning capability criterion in DPCap/DynCap analysis for marine vessels. *Ocean Engineering*, 108:164–172, 2015.

[145] S.J. Xu, D.F. Han, and Q.W. Ma. Hydrodynamic forces and moments acting on a remotely operate vehicle with an asymmetric shape moving in a vertical plane. *European Journal of Mechanics, B/Fluids*, 54:1–9, 2015.

[146] H. Zhang, Y.-R. Xu, and H.-P. Cai. Using CFD software to calculate hydrodynamic coefficients. *Journal of Marine Science and Application*, 9(2):149–155, 2010.

[147] S.A. Zuluaga, S. Rúa, R.E. Vásquez, C.A. Zuluaga, and J.C. Correa. Development and implementation of a low-level control system for the underwater remotely operated vehicle Visor3. In *OCEANS 2016 MTS/IEEE Monterey, OCE 2016*, 2016.

Appendix A: Maths

VECTOR DERIVATIVES

The derivatives herein are Jacobians. Vectors are in $\mathbb{R}^3$.

The magnitude (Euclidean norm) of $\mathbf{x} \in \mathbb{R}^3$ is computed as

$$||\mathbf{x}|| = \sqrt{x^2 + y^2 + z^2},$$

where $\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$. The unitary vector of $\mathbf{x}$ is computed as

$$\mathbf{e}_t = \frac{\mathbf{x}}{||\mathbf{x}||},$$

only if $||\mathbf{x}|| \neq \mathbf{0}$.

The derivative of $||\mathbf{x}||$ with respect to $\mathbf{x}$ is

$$\frac{\partial ||\mathbf{x}||}{\partial \mathbf{x}} = \mathbf{e}_t^\mathsf{T}.$$

If $\mathbf{r} \in \mathbb{R}^3$ is an arbitrary vector, non dependent on $\mathbf{x}$, the following derivative may be computed

$$\frac{\partial ||\mathbf{x}||\mathbf{r}}{\partial \mathbf{x}} = \mathbf{r}\mathbf{e}_t^\mathsf{T}.$$

The derivative of $\mathbf{e}_t$ with respect to $\mathbf{x}$ is

$$\frac{\partial \mathbf{e}_t}{\partial \mathbf{x}} = \frac{1}{||\mathbf{x}||} \left(\mathbf{I} - \mathbf{e}_t \mathbf{e}_t^\mathsf{T} \right).$$

For an arbitrary vector $\mathbf{v} \in \mathbb{R}^3$, its projection over an unitary vector $\mathbf{e}_t$ is

$$v_t = \mathbf{v}^\mathsf{T} \mathbf{e}_t$$

and the component over the direction $\mathbf{e}_t$ is

$$\mathbf{v}_t = (\mathbf{v}^\mathsf{T}\mathbf{e}_t)\mathbf{e}_t.$$

The derivatives with respect to $\mathbf{x}$, assuming $\mathbf{v}$ is independent of $\mathbf{x}$, are

$$\frac{\partial v_t}{\partial \mathbf{x}} = \frac{1}{||\mathbf{x}||}\mathbf{v}^\mathsf{T}\left(\mathbf{I} - \mathbf{e}_t\mathbf{e}_t^\mathsf{T}\right) \quad \text{and} \quad \frac{\partial \mathbf{v}_t}{\partial \mathbf{x}} = \frac{1}{||\mathbf{x}||}\left(v_t\mathbf{I} + \mathbf{e}_t\mathbf{v}^\mathsf{T}\right)\left(\mathbf{I} - \mathbf{e}_t\mathbf{e}_t^\mathsf{T}\right).$$

The derivatives with respect to $\mathbf{v}$ are

$$\frac{\partial v_t}{\partial \mathbf{v}} = \mathbf{e}_t^\mathsf{T} \quad \text{and} \quad \frac{\partial \mathbf{v}_t}{\partial \mathbf{v}} = \mathbf{e}_t\mathbf{e}_t^\mathsf{T}.$$

Another useful derivative is

$$\frac{\partial ||\mathbf{x}||\mathbf{x}}{\partial \mathbf{x}} = ||\mathbf{x}||\mathbf{I} + \mathbf{x}\mathbf{e}_t^\mathsf{T}.$$

INTEGRALS OF LINEAR LAGRANGE POLYNOMIALS

The basis functions of linear Lagrange polynomials are given by

$$\begin{aligned} \mathrm{N}_1^j(s) =& \frac{s_{j+1} - s}{s_{j+1} - s_j} = \frac{s_{j+1} - s}{\Delta s_j} \quad \text{and} \\ \mathrm{N}_2^j(s) =& \frac{s - s_j}{s_{j+1} - s_j} = \frac{s - s_j}{\Delta s_j}. \end{aligned}$$

They are useful to define linear functions for $s \in [s_j, s_{j+1}]$.

Basis function derivatives are given by

$$\mathrm{N}_1'^j(s) = -\frac{1}{\Delta s_j} \quad \text{and} \quad \mathrm{N}_2'^j(s) = \frac{1}{\Delta s_j}.$$

Useful integrals are

$$\int_{s_j}^{s_{j+1}} \mathrm{N}_1^j(s)\mathrm{d}s = \int_{s_j}^{s_{j+1}} \mathrm{N}_2^j(s)\mathrm{d}s = \frac{1}{2}\Delta s_j,$$

$$\int_{s_j}^{s_{j+1}} \left[\mathrm{N}_1^j(s)\right]^2 \mathrm{d}s = \int_{s_j}^{s_{j+1}} \left[\mathrm{N}_2^j(s)\right]^2 \mathrm{d}s = \frac{1}{3}\Delta s_j, \quad \int_{s_j}^{s_{j+1}} \mathrm{N}_1^j(s)\mathrm{N}_2^j(s)\mathrm{d}s = \frac{1}{6}\Delta s_j,$$

$$\int_{s_j}^{s_{j+1}} \mathrm{N}_1'^j(s)\mathrm{N}_1^j(s)\mathrm{d}s = \int_{s_j}^{s_{j+1}} \mathrm{N}_1'^j(s)\mathrm{N}_2^j(s)\mathrm{d}s = -\frac{1}{2},$$

$$\int_{s_j}^{s_{j+1}} \mathrm{N}_2'^j(s)\mathrm{N}_1^j(s)\mathrm{d}s = \int_{s_j}^{s_{j+1}} \mathrm{N}_2'^j(s)\mathrm{N}_2^j(s)\mathrm{d}s = \frac{1}{2},$$

$$\int_{s_j}^{s_{j+1}} [\mathrm{N}_1^j(s)]^2 \mathrm{N}_1'^j(s)\mathrm{d}s = \int_{s_j}^{s_{j+1}} [\mathrm{N}_2^j(s)]^2 \mathrm{N}_1'^j(s)\mathrm{d}s = -\frac{1}{3},$$

$$\int_{s_j}^{s_{j+1}} [\mathrm{N}_1^j(s)]^2 \mathrm{N}_2'^j(s)\mathrm{d}s = \int_{s_j}^{s_{j+1}} [\mathrm{N}_2^j(s)]^2 \mathrm{N}_2'^j(s)\mathrm{d}s = \frac{1}{3},$$

$$\int_{s_j}^{s_{j+1}} \mathrm{N}_1^j(s)\mathrm{N}_2^j(s)\mathrm{N}_1'^j(s)\mathrm{d}s = -\frac{1}{6}, \quad \text{and} \quad \int_{s_j}^{s_{j+1}} \mathrm{N}_1^j(s)\mathrm{N}_2^j(s)\mathrm{N}_2'^j(s)\mathrm{d}s = \frac{1}{6}.$$