An AUV Systems Model Predictive Control Approach

Rui Gomes

Electrical Engineering Department

Porto University

A thesis submitted for the degree of

PhilosophiæDoctor (PhD)

2018 September

Abstract

This thesis concerns a novel control framework of the Model Predictive Control (MPC) - designated by attainable Set-MPC - type that seeks to conciliate performance optimization and state feedback control under very strict on-line computational constraints. The key novel idea that pervades throughout the main contributions of the thesis consists in transferring very substantial computational burden associated with the building blocks of the conventional MPC scheme to the off-line stage, by taking advantage of the time invariance of fundamental subsystems.

The challenges of controlling single AUV and formations of AUV systems served as a key anchor to inspire, develop and illustrate the contributions of the thesis. The main reasons to choose this class of applications are manifold. The underwater environment is complex and poses tremendous challenges for the design of advanced data gathering systems. Space (required for payload and other devices), and energy (required for the actuation, sensing, computation, and communication) are at a premium and communications, typically merely acoustic, are difficult due to very low data rates, unreliability and power hungry. This makes the case extremely efficient management of onboard resources and this implies the need of optimization in a context of a state feedback control.

The MPC framework suits these requirements. However, it suffers from the drawback of requiring intensive computation - inherent to solving optimal control problems - in real-time. Hence, the relevance of the research undertaken in this thesis.

Besides the necessary contextual items, notably the problem statement, challenges analysis, and a directed and commented state-of-the-art review, this thesis includes an in depth assessment of the application of conventional

MPC scheme to a simple AUV formation control scenario that encompassed not only software simulation but also hardware-in-the-loop with field data context.

Based on the assessment of the application of the conventional MPC scheme, the AS-MPC scheme was developed. This requires the off-line computation of the Attainable Set and of the system Value Function to be adapted in the on-line context with a very small computational effort. Results on asymptotic optimality, and asymptotic stability, required to formally ensure the desired properties of the AS-MPC scheme were proved. Moreover, a discussion on robustness and computational tractability and the migration of some conclusions and results from conventional MPC schemes to the AS-MPC scheme was included, giving rise to the Robust AS-MPC (RAS-MPC) also developed in the context o this thesis. Given the complexity (even in the off-line stage) of computing Attainable Sets and Value Functions, in this thesis we proposed a novel approach to approximate these sets through a cloud of points with the suitable properties.

Finally, given the hybrid - that is, discrete event and continuum-time driven - nature of the envisaged class of systems, this thesis also includes an analysis of critical issues arising in this context. Now, even for the AS-MPC scheme, there is a lot of on-line computational effort that cannot be transferred to the off-line stage. By resorting to well-established Process Systems Engineering methodologies, an accurate as possible hybrid control system is developed whose a priori decoupling of discrete-event and continuum time components enables to represent the overall system through an hybrid automaton that will provide the controlled dynamics (in a hybrid systems sense) to the AS-MPC (or, obviously, RAS-MPC). The resulting control architecture is explained through illustrative examples related to motion control and obstacle collision avoidance activities.

Finally, a number of conclusions and open issues that emerged from the research effort underlying this thesis are presented and discussed.

A man ought to read
*j*ust as his inclination leads him;
*f*or what he reads as a task
*w*ill do him little good.

Samuel Johnson

Acknowledgements

I would like to express my sincere gratitude to everyone that, directly or indirectly contributed to the development of this thesis.

First of all, I would like to thank my advisors Prof. Fernando Lobo Pereira and Prof. João Tasso Borges de Sousa for their support and guidance during my stay at the Underwater Systems and Technology Laboratory (LSTS). They had not only provided professional guidance as inspiring mentors but also the resources to make this work possible.

Secondly, I would like to express my deepest gratitude and special thanks to Prof. Anthony Healey, whom I worked with on topics related to this thesis. Many thanks for supporting my stay at the Naval Postgraduate School in Monterey, CA.

I also want to thank all my colleagues whom I interacted with, specially the LSTS people: Sujit, Fernando Auat, Paulo Dias, Pedro Calado, Ricardo Martins, Renato Caldas, João Pereira, Bruno Loureiro, Joel Gomes, José Pinto, Rui Gonçaves, Eduardo Marques, Márcio Correia, Alexandre Sousa and Luis Madureira, who have been instrumental in creating a friendly and intellectually exciting atmosphere in the lab.

I am also thankful to *Fundação para a Ciência e Tecnologia* (FCT) for the financial support, to the Faculty of Engineering of Porto University for being such a good school and for everything that I have learned throughout the years.

I can not forget Sérgio Fraga, Berta Estevinho, Hossein Shahzamanian, Salomé Teixeira and Helena Branco for years of fruitful discussions, friendship and priceless advice.

Finally, I thank my family for their love and constant support in my endeavors and encouragement over the years.

This work had the financial support of POPH-QREN in the form of PhD scholarship: SFRH/BD/19179/2004/6950.

Contents

List of Figures

List of Tables

Glossary

AOCP Auxiliary Optimal Control Problem

APDL Administração dos Portos do Douro, Leixões e Viana do Castelo, the local Porto harbor

ARV Autonomous Robotic Vehicle

AS-MPC Attainable Set Model Predictive Control

ASV Autonomous Surface Vehicle

AUV Autonomous Underwater Vehicle

C4C Control for Coordination of Distributed Systems FP7 Project

CLF Control Lyapunov Functions

CTD Conductivity, Temperature and Depth

DUNE Uniform Navigation Environment software developed at the Underwater Systems and Technology Laboratory

DVL Doppler Velocity Logger

GPS Global Positioning System

GSM Global System for Mobile communications

HIL Hardware in the Loop

HJE Hamilton Jacobi Equations

IMU Inertial Motion Unit

IQC Integral Quadratic Constraints

LBL Long BaseLine

LQOCP Linear Quadratic Optimal Control Problem

LSTS Laboratório de Sistemas e Tecnologia Subaquática (Underwater Systems and Technologies Laboratory from Porto University)

MLD Mixed Logical Dynamical

MPC Model Predictive Control

NEPTUS Mission planning, control and post mission analysis software console for all vehicles available at the Underwater Systems and Technology Laboratory

NMPC Nonlinear Model Predictive Control

OCP Optimal Control Problem

PMP Pontryagin Maximum Principle

PWA Piece-Wise Affine

RAS-MPC Robust Attainable Set Model Predictive Control

RHC Receding Horizon Control

ROV Remotely Operated Vehicle

UAV Unmanned Air Vehicle

WP Work Package

Chapter 1

Introduction

This thesis concerns a novel Model Predictive Control (MPC) approach to address a very general class of systems for which some sort of approximation to optimal control strategies is of interest while satisfying constraints on available resources, such as, time, space, power, computation, communications, sensing, among others, required to perform the activities underlying their specified operations. In many instances, these constraints can be so severe that either the performance requirements, or, even, the viability of the operational success - accomplishment of the system's purpose - of the system require an optimized trade-off of the distribution of the resources consumption among the multiple activities necessary for the system's operation.

There is a long history of developments on MPC, (1). However, the tremendous success has been achieved for systems whose requirements are not so hard. In general, the dynamics of the system are very slow relatively to the computational speed, the availability of data is not a major concern, and the power required for computation is not an issue. In this thesis, we consider the development of MPC schemes for the control problems of motion - navigation, guidance, and control - and of other activities that exhibit requirements that contrast starkly with those of this "easy" scenario.

Given the current technological state-of-the-art, systems of networked Autonomous Robotic Vehicles (ARVs), and, with particular emphasis for Autonomous Underwater Vehicles (AUVs), fall in the class of systems for which it is extremely important to optimize the consumption of on-board resources in order to ensure the value of the system's operation. Typically, this involves finding a trade-off between endurance, quality-of-service or amount of gathered information, precision of navigation and motion control, and extent of communications for payload data transmission or required for navigation and control activities underlying the system's mission.

Thus, although this thesis focus on a general MPC scheme, we consider the single or multiple AUV motion control problem as a paradigmatic general case study, that has been addressed in works such as (2) and references therein, for the following main reasons:

- It contains all the important the ingredients that serve as an inspiration and motivation to develop the MPC approach satisfying very general requirements encompassing a very wide range of applications.
- The fact that LSTS (http://lsts.fe.up.pt/) made available its infra-structure that provides access to a rich operational experience with such systems, as well as the potential to test the practical viability of the developed MPC approach.
- Systems based on the coordination of multiple AUVs and, possibly, also other type of robotic vehicles and other systems constitute an extremely promising and powerful source of inspiration for conceptual and applied developments that will play a role in addressing the key challenges that human kind is facing today.

Before we pursue with the rationale of this thesis, let us dwell on the last motivation item underlying AUV based systems. It is a fact that the marine environment is one of the wealthiest sources of data necessary to understand and interact with most, if not all, the natural phenomena underlying the extremely critical real world challenges that human kind is perceiving today. Challenges associated with climate changes, biodiversity, environment, natural resources management, territory management, security and surveillance, to name just a few, impose a number of increasingly sophisticated requirements for field studies data gathering. Spatial and temporal distribution, persistence, combination of wide area with local area data sampling, etc., are some of the general requirements calling for a concerted instrumentation of the earth which encompasses fixed, mobile sensor platforms, and other devices networked as required by the evolution and emergence of the needed knowledge. Moreover, one can easily devise many instances of missions involving, possibly heterogeneous, networked unmanned vehicles, say AUVs, or other ARVs, and a huge number of diverse devices, in which there is the need to distribute different sensors by different vehicles that should move in a certain formation defined to fulfill the specified data sampling requirements.

The huge difficulties involved in the extraction of data from marine environments - communications, increasing pressure with depth, relative chemical and physical aggressiveness - make it particularly important the design of devices that can execute pre-planned missions or, even better, are endowed with some deliberative capabilities. These are of utmost relevance due to the combination of high variability to be encountered and the extremely practical difficulty of human intervention during the execution of missions. It is not hard to imagine the tremendous complexity that the design of this class of systems entails.

In this thesis, we are just concerned with the small building block of designing systems enabling the optimized motion control of one or more AUVs in the context of the execution of complex missions. The nuclear idea on which the set of contributions of this thesis relies consists essentially in de-conflicting requirements inherent to the computational complexity generally associated with typical MPC schemes and those associated with the satisfaction of very strict constraints on time, power, and computation that emerges in the real-time running this class of marine systems.

The fact that typical MPC schemes involve solving an optimal control problem for a relatively long planning horizon in each relatively small control time interval after the state variable sampling, makes their computational complexity very high and difficult to conciliate with real-time constraints. Even if reasonably simplified models are used, (2), data has to be gathered - either by communication or by sampling sensed data -, and several optimization problems - always involving a considerable number of variables and of constraints -, have to be solved in a very short time interval to generate the control signals to be applied to the actuators low level controllers in the short control time horizon. Moreover, it is often the case that uncertainty has to be handled by running estimation procedures on the available data. It is a significant challenge to ensure that time constraints are met with a low power budget and with a typically small onboard processor.

The key idea to tackle this challenge consists in, by taking advantage of time-invariant data underlying the formulation of the involved optimization problems, computing a priori (i.e., off-line) a number of, as comprehensive as possible, simple building blocks required for optimal or sub-optimal control synthesis as a function of a set of parameters associated with a satisfactory number of more likely typified on-line situations. This, possibly large, amount of data is stored on onboard in appropriate look-up tables. Then, the actual on-line control synthesis is performed by computing the parameter values that take into account the sampled data to retrieve the best control values from the look-up table. The scheme is such that the computation involved in the on-line adaptivity requires several orders of magnitude smaller time and computational power relatively to the conventional MPC schemes.

1.1 General challenges

The underwater milieu is a complex and difficult environment that poses tremendous challenges for the design of advanced data gathering systems such as the one in which the motion control and guidance systems envisaged in this thesis will play a role. Space - required for payload and other devices -, and energy - required for the actuation, sensing, computation, and communication - are at a premium and communications, typically merely acoustic, are difficult due to very low data rates, unreliability and high power budget. This makes the case for the careful management optimization of onboard resources in their allocation to the various subsystems in order to perform the activities to accomplish the mission objectives.

Moreover, hydrodynamic phenomena affects the vehicle performance. The precise models are too complex from the computational point of view and needs to be approximated by simpler concentrated parameter models. The price to pay for this is that modeling becomes more difficult and uncertainty increases. Uncertainty is aggravated by the impact of typical underwater perturbations which are significant and pervasive in that they affect all the subsystems, specially those pertaining to navigation and control.

Missions that need persistent spatial and temporal sampling of phenomena are of key importance for ocean studies. For such missions, we need a set of sensors that can sample the data as a given phenomenon develops which is dynamic in nature. To fulfill the specific sampling requirements, a set of one or more AUVs and, possibly other devices, must be deployed in the region of interest in order to move, communicate, and perform payload activities in a coordinated fashion to achieve the mission goals in spite of the, often significant, multiple environmental disturbances and technological constraints.

This requires the ability of controlling the robotic vehicles in such a way that specific, often complex, and, possibly varying in the course of the mission, space and time position constraints are satisfied by one or more vehicles. Thus, a very versatile motion control synthesis is required. Moreover, these requirements have to be satisfied while optimizing on-board resources, notably, power consumption. This is a sophisticated challenge since, depending on the mission specification, it is generally hard to formulate an optimization problem whose solution reflects the best trade-off of the available power allocation to multiple activities: actuators, sensing (payload and navigation data gathering), communication (payload and navigation data transmission and reception), and information processing (payload, navigation and control). One should add the fact that the system does not operate in a deterministic context. This means that, in order to best achieve the mission goals, a substantial change in the policy for the usage of on-board resources may take place depending on the occurrence of discrete events such as device failures, unpredicted environment changes, as well as unexpected features of the phenomenon of interest.

Clearly, the overall problem constitutes a formidable challenge. However, we will focus on the isolated problem of motion control (at various levels of abstraction) as it plays a key role in the overall system operation. The much simpler problem of controlling the motion of an AUV or a set of AUVs during the execution of a mission in a well defined context while optimizing the on-board resources poses in itself significant challenges and constitutes a prime building block in the design of the overall system. This is the main single challenge in which this thesis is focused.

1.2 Objectives

In this section, we start by describing the context in which the objectives of this thesis will be stated.

In this thesis, we adopt the optimal control problem with both control constraints and state constraints as the framework to formulate the general problem of controlling the motion of a set of one or more AUVs to achieve the task of collecting spatial-temporal data with certain geometric constraints between the measurements in coordinated fashion. State constraints can be quite versatile and reflect the possibility of not only considering, possibly time-varying, AUVs formation patterns, but also, including collision avoidance with unmapped obstacles if a state feedback context is used.

As it was mentioned earlier, the operational environments of AUV systems are usually plagued with uncertainty and high variability. This means that state feedback control strategies should be adopted in order to take into account not only the state of vehicle being controlled but also that of those with which it interacts as well as the "state" of the environment. For this purpose, the control synthesis based on optimal control is embedded in a receding horizon - the so called MPC - scheme in which the pertinent fraction of the overall "state" is sampled at a certain points in time that will be considered the initial state from which the optimal strategy is computed for the next optimization horizon. Remark that the sequence of sampling times may depend critically on the characteristics of the pertinent on-board sensors, and their specification can be part of the closed loop control strategy.

Moreover, vehicle formations can be controlled through centralized or decentralized controllers. The centralized controller requires the state information from the vehicles to compute control inputs for the next time step. In ocean waters with low acoustic bandwidth and delays it is not possible to achieve the information on demand. Hence, decentralized controllers are preferred that can compute the desired control independently with delayed information feedback arriving asynchronously from other vehicles. This type of formation control problem is also referred as "leaderless formation control" problem. In this approach, a group of autonomous vehicles travel along a predefined trajectory while maintaining a desired spatial pattern and connectivity. Each vehicle has its own onboard sensing, computation, and communication capabilities. When submerged, the AUVs have, in general, limited communication capabilities, and, thus, not all the global information is available to each one of them. The design of the overall control system for each vehicle has to be based on the local information. Since no leader is designated, all robots will have to coordinate with each other by relying on some global consensus for a common goal achievement.

The three main classes of objectives of this thesis are as follows:

- Assessment of the conventional MPC scheme to control a formation of AUVs.

Here, we consider not only the issues of computational complexity, and communications reliability but also their impact in the overall system performance which, naturally, encompasses the AUVs navigation and control, for the cases of centralized, decentralized, and incorporation of obstacle collision avoidance schemes.

Computational complexity is of paramount importance. The limited on-board computational resources coupled with the required sampling times leads to the consideration of simple models. In Chapter 4, we consider discrete time linearized models of the AUV dynamics since, in this case, the procedure to solve the optimal control could be reduced to that of a large linear quadratic problem for which there are extremely efficiently solvers which can be effectively solve on-line the formulated optimization problems underlying the control synthesis. On the other hand, difficulties in obtaining the desired performance are encountered when decentralized conventional MPC schemes are implemented. One of the most important factor in decentralized formation control, particularly, in the presence of unreliable communications is to have a very good AUV model. In the absence of significant perturbations, this is important to generate accurate trajectory predictions for formation control synthesis both in simulation and in real-world experiments. The parameters of the models must be estimated using parameter identification techniques. In this thesis, we devise a simplistic but efficient approach in developing good models for underwater vehicles along with parameter identification. With the devised model of the underwater vehicle, we should develop a robust formation controller that can handle currents and delays in feedback information about the state of other vehicles. Our approach in dealing with this scenario is given in Chapter 4.

- Specification and investigation of an Attainable Set MPC scheme and issues inherent to its implementation.

  This objective was motivated by the straightforward observation that for large classes of time invariant systems, the conventional MPC schemes involve a quite large number of repetitive computationally demanding optimization processes - in which the integration of system dynamics has a significant role - in very similar circumstances with often irrelevant differences. These prompts the following general pertinent question: why not to pre-compute the dynamics for a number of controls as a function of a certain number of parameters, store the results in a look-up table, and, then recruit them in real-time according to the prevailing conditions determined by sensed or communicated data in order to take into account these small perturbations?

The proposed Attainable Set implementation of the MPC scheme is based on the assumption that the vehicle dynamics are time-invariant. The basic idea is to replace the procedure of solving the optimal control problem - which in continuum time is infinite dimensional - by an a priori much simpler problem in which, at each step the equivalent to the overall cost functional provided by the Value Function is optimized on the set of points of the state space that can be attained at the final time of the control horizon. Both the Attainable Set and the Value Function are, then "adjusted" in real-time according to the received data (navigation-pertinent, and payload events of interest) in the course of the mission execution.

- Integration of the Attainable Set implementation of the MPC scheme in a Control Architecture.

  In the previous item, the real-time "adjustment" of the pre-computed Attainable Set and Value Function were considered merely in the context of small perturbations. Thus, the natural question arises: if unexpected events with a very significant impact in the system behaviour occur, is the proposed MPC scheme still useful in efficiently determining near optimal control strategies?

  The answer is positive by considering a hybrid model for the dynamics in which the multiple - reasonably exhaustive - typified modes of operation are included. These might imply different navigation schemes and/or controllers (e.g., sensor-based motion, a denser set of way-points, control in position or in velocity, etc.), changes in the formation configuration, etc.. It is shown that, although higher than that of the previous item, under a reasonable circumstances, the on-line computational burden associated with the online "adjustment" of the Attainable Set and the Value Function is still manageable by the on-board computational resources.

1.3 Approach

Obviously, different approaches were adopted in order to pursue the objectives of this thesis stated above.

In order to assess the performance and limitations of the conventional MPC scheme to control a formation of AUVs, the following approaches were considered and associated activities were undertaken:

1) Characterization of formation control challenges and problem formulation, including natural and technological constraints and specification of requirements. This leads to the identification of design challenges and constraints (control, computation, communications) to be dealt with by the overall control design.

2) Examination of the current state-of-the-art results and technologies in control that best meet the identified challenges, and investigation of further developments, at both the scientific and technological levels, required to enhance the performance and to ensure the requirements of the overall system.

3) Development of a specific simulation framework to test the designed control system. This will be used to assess the performance of the overall system but also to fine tune the design in order to exploit all possible performance improvements while meeting the desired targets.

4) Migration of the designed control system from the simulation context to the AUVs' systems in such a way as to take into account the vehicle specific features as well as the integration of the control system in the control architecture. Testing of the control system in four scenarios of incremental complexity.

In the formation control literature, there are wide variety of control approaches that can be used to achieve and maintain the desired formation. A brief outline of the current state of the art on formation control is given in Chapter 2. Since the MPC method to design the formation controller will be used in this thesis, a special attention is paid to this class of controllers.

The MPC methodology allows one to control a dynamic system by combining prediction and control. The plant model provides the trajectory prediction for a control function computed to ensure the desired performance. Then, this control is applied during a predetermined period of time, and, once this period elapsed, the state of the system is sampled and the actual state is compared with the predicted one. At this point in time a new control strategy is computed in order to compensate for the detected "error" in order to achieve the desired objectives while respecting the plant's constraints. Such constraints include the actuators physical limits and boundaries of safe operation. Due to this prediction and control phenomena, it suits the AUV formation control problem as this problem has delays in acquiring state information about other vehicles. During this delayed period, the AUV can predict the potential trajectory of the neighbors and generate the desired control trajectory. Details on the formulation of the MPC problem and its advancements to reduce computational overhead using Attainable Sets[1] is the core contribution of this thesis.

In what concerns the "Specification and investigation of an Attainable Set implementation of the MPC scheme", the ingredients of the overall approach involves the following items:

1 Equivalence between the conventional Optimal Control Problem (OCP) and a certain Finite Dimensional Nonlinear Optimization Problem (FDNOP).

2 Efficient approximations of the Attainable Set and of the Value Function for a given Optimal Control Problem.

3 A simple reformulation of the conventional MPC scheme as a time sequence of certain FDNOPs.

4 Investigation of the properties of the scheme devised in the previous item.

5 A robust version of the scheme devised in item 3.

The first item requires that the data of the optimal control problem satisfies the principle of optimality. That is, at any point $(t, x) \in [t_0, \infty) \times I\!R^n$, the Value Function can be obtained by solving the Hamilton-Jacobi-Bellman equation. The huge computational complexity in solving this equation is well-known. However, these computations are performed off-line and, moreover, the associated burden can be further mitigated by

[1] Some works in this area make no distinction between Attainable Sets and Reachable Sets. Even though the term Reachable Set was used in previous publications related to this work, we actually mean Attainable Sets.

a proper choice of simplified models adapted to the specific modes of operation. For the reader unfamiliar with these concepts, and how the associated objects can b computed and handle, we point out to chapter 5 of this thesis, where specialized references will be indicated. We remark that the essence of the concepts, constructions, and implementations associated with these objects is independent of whether a single vehicle or an, either centralized or decentralized, formation of vehicles are being addressed. Of course, the same does not happen to the case of the overall system control design.

In what concerns the “Integration of the Attainable Set implementation of the MPC scheme in a Control Architecture”, the ingredients of the overall approach involves the following items:

1 Discussion of issues inherent to the context in which the state of the system is steered by both the usual continuum-time control strategies, and occurrence of either controlled or uncontrolled discrete events. This prompts the formulation of the mission execution control as a general control problem whose dynamics are formalized by the so-called controlled hybrid dynamic systems. Hybrid Automaton provides one of the most popular modelling framework.

2 The occurrence of unexpected significant events brings about the need to specify a set of diverse various modes of operation - which may require different models, constraints and performance measures - as well as the need to specify the various events that trigger the transition from one mode operation to another.

3 It follows from the previous item, that significant “adjustments” are required in both the Value Function and the Attainable Set for the different modes of operation as well as to the on-line schemes the needed real-time adaptivity.

4 The implementation of the Attainable Set MPC encompasses the formulation of optimal control problems whose dynamics are given by Controlled Hybrid Systems, and the new level of complexity requires the specification of a Control Architecture to facilitate the implementation of the MPC scheme.

5 In a first instance, the previous item will be addressed in the context of a single AUV, and, afterwards, it will be extended to the case of the hybrid systems control of a formation of vehicles.

1.4 Thesis contributions

The contributions of this thesis are solidly grounded in experimental work -being the initial phase undertaken under the EU FP7 research project “C4C - Control for Coordination”, FP7-ICT-223844.

1. Development of a conventional MPC distributed control test framework. This encompassed also Modeling of AUV for parameter estimation and AUV formation control with acoustic feedback. This work has been published in (2, 3, 4).

2. Design and development of an Attainable Set MPC (AS-MPC) scheme to deal with computational complexity problem. This allows real-time requirements to be fulfilled for some application. This work has been published in (5).

3. Design and development of RAS-MPC, a version of the AS-MPC scheme in the previous item which is robust to low intensity but persistent perturbations. This work has been published in (6).

4. Integration of the of the RAS-MPC in the control architecture by setting up a hybrid RAS-MPC. This work has been published in (7, 8).

1.5 Organization of the thesis

This work is organized as follows. Chapter 2 discusses in detail the various ingredients in the formulation of the AUV formation control problem and justify the options made in the adopted formulation. Also an overview of the state-of-the-art on model predictive control will be given in Chapter 3.

Since AUV models are, in general, extremely complex to obtain, and, then, to work with, mainly in the real-time context, modeling simplification, and parameter identification methods are discussed, being the more standard material placed in Appendix C. Since different modes of operation require different models, the content of this chapter will be useful to address the coordination problem arising in the MPC scheme implementation to control multiple vehicles.

The approach adopted to address the decentralized formation control problem is presented in Chapter 4. This includes both the research issues being tackled, the control framework design methods, as well as the systems engineering process leading to the future implementation of resulting formation control framework. All issues about the migration of the simulated MPC framework to the real vehicles environment is presented in Chapter 4. Several Hardware In the Loop (HIL) simulations and their analysis are also included.

In Chapter 5, the Attainable Set formulation of MPC will be presented and discussed. This includes issues of asymptotic stability, optimality, robustness, and of computational tractability. Results supporting the properties of several of the required constructions will be presented and proved. Simulation results for both single and multiple vehicle cases will be presented.

The main body of the thesis is closed in Chapter 7 with the main conclusions concerning the implementation of the Attainable Set MPC control framework in general and for AUVs in particular. A certain comparison with conventional MPC implementations will also be done, specially for AUV applications. Current open challenges and prospective future work will also be discussed in this last chapter.

Finally some auxiliary conventional material will be included in the appendices.

Chapter 2

Autonomous Underwater Vehicle Formation Control Problem

2.1 Introduction

In this chapter, we start by presenting a formulation of a general problem of controlling a formation of robotic vehicles in order to track a given path which might be pre-defined or specified by another moving vehicle, and, then, we will focus on a survey on some of the state-of-the-art on the control of formations of Autonomous Robotic Vehicles (ARVs) with some emphasis placed in underwater vehicles (AUVs).

Before pursuing that, it is worth to discuss the reasons behind the increasing interest in the control of multiple vehicles as well as, at least outlining, the wide variety of approaches to address this general problem. This will be done in the next section.

Another reason to expose the diversity and complexity of issues arising in the control of single and in multiple AUVs consists in strengthening the case for the full development of the novel approach for the Attainable Set MPC schemes that will be presented and discussed in Chapter 4.

2.2 Why multiple vehicle control problems and general approaches

From the systemic point of view, it is clear that the degree of sophistication and diversity of missions, reliability and robustness in their successful completion are clear advantages of multiple ARV systems relatively to single vehicle systems. Moreover, the overall achievable performance - quality of service, execution time, robustness, effectiveness and safety - in reaching the desired goal of multiple ARVs systems is much superior than that of a single vehicle operating in the same environment under the same conditions. Versatility in the execution enabled by task allocation, exchange of roles, distribution of payload navigation devices among the various ARVs are some of the key reasons. See (9, 10, 11, 12).

Systems of multiple ARVs may be of very diverse nature: heterogeneity (ground, aerial, space, underwater, or marine surface vehicles as well as industrial manipulators, biped, and exoskeletons), roles (homogeneous or heterogeneous), and interaction relationships (leader-follower, leaderless, cooperative, coordinated, hierarchic or heterarchic). The classes of applications are also extremely diverse: surveillance (13), exploration (14), communications, remote and in-situ data gathering: via satellite clustering (15), networks of underwater autonomous vehicles (16, 17, 18, 19), aerial vehicles and unmanned aerial vehicles (UAVs) (20, 21), (22), cooperative robot reconnaissance (23), and manipulation cooperation (24). In such applications, multiple robots are required to travel autonomously between different locations, while avoiding collisions with static or dynamic obstacles and other robots, even physical faults occurred at individual member of the team or communication between members of the team.

Approaches to the control of multiple ARVs can be organized in three grand classes: centralized, decentralized, and mixed centralized-decentralized systems. In a centralized system, a powerful core unit makes decisions and communicates within the vehicles in the team. This core unit can optimize vehicle coordination, accommodate individual vehicle faults and monitor the accomplishment of the mission. However, it is possible that any faults in the core can facilitate a failure of the whole system. Moreover, Centralized approaches do not scale well as formation size increases, do not utilize the computational resources available on each vehicle, and incur in large communication overhead. This is true even when the most advanced optimization solvers are used.

In the decentralized approach, which is in part inspired by the social aggregation phenomena in birds and fish (25, 26), each vehicle can communicate and share information. Clearly, to extract the maximal capabilities from decentralized schemes, more sophisticated control and communication schemes are required to overcome the limitations of a priori allocation of specific tasks that limits considerably the overall potential performance. Decentralized systems optimize the allocation of resources, so vehicle faults can also be overlooked, but this can result in a less efficient mission outcome. Decentralized systems are less affected by computational and communication bottlenecks, and they are more robust to the loss of individual member in the team than the centralized ones. Other advantages in decentralized systems such as robustness to single agent failures, scalability of the system, time constraints of applications, constraints on communication load, and computational power of the agents. System-level cognitive operations, though, are much more difficult to implement in decentralized systems.

Different architectures and strategies have been developed in either centralized or decentralized methods in order to control and coordinate a multiple ARVs group. These may be organized in the following - obviously non-disjoint - groups: behavior-based (23, 27), virtual structure (21, 28, 29, 30), leader-follower (31, 32), graph-based (33) and potential field approaches (34), and combinations of these in order to obtain the desired requirements.

- *Virtual Structure.* The entire formation is treated as a single entity, and the desired motion is assigned to the virtual structure that traces out the trajectory for each member of the formation to follow. Behavior coordination for a group of mobile robots in virtual structure approach is uncomplicated and that is the advantage of this method. The disadvantage of the current virtual structure implementation is the centralization, which leads a single point of failure for the whole system. In (28) formation control ideas for multiple spacecraft using virtual structure approach are presented.

- *Behavior Based Methods.* In this approach, several behaviors are available to each robot and the final control is derived from a weighting of the relative importance of each behavior, but there is lack of modeling for the subsystems or robot surroundings. In (23), reactive behavior-based approach is introduced that implement formations integrated with navigational behaviors to enable a robotic team to reach navigational goals, avoid hazards and simultaneously remain in formation. Lawton (27) presents a behavior-based approach to formation maneuvers for groups of mobile robots. Complex formation maneuvers are decomposed into a sequence of maneuvers between formation patterns. Hardware implementations illustrate the effectiveness of the proposed control strategies. However, it is not easy to ensure the required group dynamics that guarantee the stability of the whole system.

- *Leader-Follower Approaches.* In this approach, one of the robots is designated as the leader, with the rest robots as followers. The follower robots need to position themselves relative to the leader and to maintain a desired relative position with respect to the leader. This method is characterized by simplicity and reliability. In this method, there is no explicit feedback from the followers to the leader and that is the disadvantage of this method.

- *Artificial Potential.* Artificial potentials define interaction control forces between neighboring vehicles and are designed to enforce a desired inter-vehicle spacing specifying the desired goals of the mission defined to the overall team of robots.

- *Graph Theoretical Approaches.* Graphs are mathematical structures that have been long used to model pair-wise relations between objects from a certain collection. A "graph" in this context refers to a collection of vertices or "nodes" - each of which may correspond to a robot - and a collection of edges that connect pairs of vertices that may define the type of interaction between robots. Some research has been done on the coordination of multiple ARVs using graph theory.

- *Intelligent Control.* This approach is based on mimicking the way that the human brain makes decisions by grouping similar objects together, and so creates faster and more accurate response times in the decision making process. For some classes of problems, this approach enables the simplification of the computations underlying the controllers. In particular, it has distinct advantages in multiple ARVs modeling, where multiple robots are moving along designated paths and simultaneously being directed with rapid velocity changes.

2.3 General formation control problem

In this thesis, we will consider the context provided by the mathematical control theory in order to formulate and design controllers for ARV systems. Moreover, we will focus on tools of dynamic systems and optimal control in order to support the synthesis of control strategies for either single or multiple ARV systems.

Let us consider the path following problem for a formation of autonomous robotic vehicles (ARVs) - each one modeled by the popular and simple unicycle - that operate in a decentralized way. By this, we mean that each vehicle defines its own control action only with its own navigation data and that from its neighbors in the context that the whole set of vehicles constitute the vertices of a connected graph.

Consider a set of n ARVs (any type of autonomous robotic vehicles) moving on a plane. For simplicity, we assume that each member has the same mechanical structure and each ARV described by the following unicycle model in global coordinates:

$$\begin{array}{l} \dot{x_i} = u_i \cos(\theta_i) \\ \dot{y_i} = u_i \sin(\theta_i) \\ \dot{\theta_i} = \omega_i \end{array} \tag{2.1}$$

where $i = 1, 2, \ldots, n$.

We consider each of the n ARVs to be a vertex of a control graph $G = (\nu, E)$, with in ν vertices and E denoting the set of edges. The pair $(j, i) \in E$ is an edge of the graph G if the state of ARV j is available to ARV i. For an undirected graph G with n ARVs the adjacency matrix $A = A(G) = (a_{i,j})$ is $n \times n$, where $a_{i,j} = 1$ if there is one edge $(j, i) \in E$, otherwise $a_{i,j} = 0$. Let N_i be a collection of neighbors of ARV i. The desired geometric formation $\mathcal{F}$ is described by the set $\{(\hat{x}_i, \hat{y}_i) : i = 1, \ldots, n\}$ for ARV i in global coordinates. The desired trajectory $\mathcal{T}$ for the formation group is described by:

$$\begin{array}{l} \dot{x_d} = u_d \cos(\theta_d) \\ \dot{y_d} = y_d \sin(\theta_d) \\ \dot{\theta_d} = \omega_d \end{array} \tag{2.2}$$

where (u_d, ω_d) are known functions of time. Our control problem is defined as follows. Formation Tracking Control Problem: Design a controller for each ARV based on its state and its neighbors' states such that the group of ARVs comes into formation $\mathcal{F}$ and the group of ARVs move along the desired trajectory $\mathcal{T}$, i.e., design control laws for systems (2.1) and (2.2) such that:

$$\lim_{t \to \infty} \left(\begin{bmatrix} x_i - x_j \\ y_i - y_j \\ \theta_i - \theta_j \end{bmatrix} - \begin{bmatrix} \hat{x}_i - \hat{x}_j \\ \hat{y}_i - \hat{y}_j \\ 0 \end{bmatrix} \right) = 0, i, j = 1..n, i \neq j, \tag{2.3}$$

and

$$\lim_{t\to\infty} \frac{1}{n} \sum_{i=1}^{n} \left(\begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} x_d \\ y_d \end{bmatrix} \right) = 0. \tag{2.4}$$

There exists a wide body of literature that address the above formation control problem. Different types of controller have been developed for various applications that include spacecrafts, unmanned aerial vehicles (UAVs), AUVs, and ground robots. The proposed specific solutions target the specific goals and the desired formation requirements. In Section 2.4, we briefly describe several types of formation control solutions depending on the type of vehicle and then focus on the development of a new controller for AUVs in this thesis.

Let us note that, due to perturbations - large or small - conflicts between tracking the planned trajectory and keeping the specified formation may arise in the motion of the ARVs. In this case, it is natural that the control seeks the best trade-off between these two goals by taking into account the specified performance measure. Another important issue concerns the fact that in a decentralised structure, each vehicle does not have full information of all other vehicles. Thus, the controller of each vehicle has to generate signals to the actuators by taking into account only its own navigation data, the one of its neighbors as well as the overall reference trajectory, in such a way that the behaviors vehicles converge to the desired ones of the global formation. The fact that, typically, uncertainties are significant and communications may fail, the overall control synthesis requires the combination of the estimation of the pose of multiples vehicles with the minimization of errors to desired formation pattern and to the reference trajectory to be tracked. Moreover, the time taken by communications and computation entails a delay that, as it is well known, constitutes a significant threat to the overall stability of the system. Clearly, real-time constraints is an important issue to take into account when devising such a control system.

2.4 Brief state-of-the-art on ARV formation control

Formation control of robotic systems has been subject of wide and intensive research and has been applied to all types of vehicles and applications.

Spacecraft formation flying is required for applications like, monitoring of the Earth and its surrounding atmosphere, geodesy, deep space imaging and exploration, and in-orbit servicing and maintenance of spacecraft. In these applications, spacecrafts need to be in tight formation to enable vehicles to act together as a single platform. Various types of controllers from non-linear control theory have been popular in this domain. Kristiansen and Nicklasson (35) present a review of existing methods on tight spacecraft formation flying that use state feedback. Lv et al., (36), developed a backstepping controller which is robust to input constraints and parameter uncertainties for spacecraft formation. Ren and Beard, (28), use virtual structure approach for formation control. Breger et al., (37), developed MPC based formation controller with sensing noise. Unlike the above approaches, Liang et al., (38), developed decentralized coordinated attitude control laws using behavior-based control approach instead of classical non-linear control techniques.

Several applications like search and rescue, agriculture coverage, security and patrol, etc, that require robot formations have also been considered. Over the years, multiple controllers have been synthesized for different applications considering non-holonomic nature vehicles and its disturbances. One of the primary formation controller was developed by Balch and Arkin, (39), where they use behavior based control to design the goals that will allow the robots to achieve formation. Most of the other formation control algorithms have been developed using leader-follower strategy. Consolini et al., (40), present a leader-follower formations of nonholonomic mobile robots, where the control inputs are forced to satisfy suitable constraints that restrict the set of leader possible paths and admissible positions of the followers with respect to the leader. Ghommam et al., (41), present a virtual structure control strategy for the coordination of multiple mobile robots using unicycle model. Other types of formation controllers using MPC and fast marching numerical methods to solve certain classes of Hamilton-Jacobi partial differential equations have also been developed. Liang et al., (42), considers the problem of formation control and obstacle avoidance for a group of nonholonomic mobile robots using MPC. Garrido et al., (43), present the application of the Voronoi Fast Marching (VFM) method to path planning of mobile formation robots. Martinez and Bricaire, (44), design a novel formation control strategy with collision avoidance for point robots moving in the plane. The control law is based on the design of attractive and repulsive vector fields which guarantee the non-existence of undesired equilibria. Mastellone et al., (45), presents a feedback law using Lyapunov-type analysis that guarantees collision avoidance and tracking of a reference trajectory for multiple nonholonomic robots maintaining a specific formation. Finally, (46), used backstepping for the formation control of mobile robots in the presence of disturbance uncertainties.

There are formations of UAVs which are different from that of ground robots and spacecrafts: The team of UAVs are required to fly in unison with tight formation control with tip to tip coverage for drag reduction that decreases fuel consumption. Also, this kind of formation allows the team to be viewed as a single aircraft by radar systems, (47). Several formation controller have been proposed by using adaptive control, (48), leader-follower control strategies, (20, 49, 50), sliding mode based controller, (51), and behavior based controller, (52).

Other proposed solutions include a nonlinear model predictive control (NMPC) framework for collision-free formation flight controller design for unmanned aerial vehicles, (53), being the formation flight controller designed in a distributed way, and a NMPC for guidance of a fixed-wing UAV in precision deep stall landing, (54). We also found applications of output-feedback MPC, (55), where the problem of two UAVs tracking an evasive moving ground vehicle is solved, and a comprehensive framework for the cooperative guidance of fleets of autonomous vehicles relying on MPC and addressing challenges as collision and obstacle avoidance, formation flying, and area exploration, (56). Deelopment of a path tracking model predictive control of a tilt-rotor UAV carrying a suspended load can be found in (57), and a solution for formation control with collision avoidance for a multi-UAV system using a decentralized MPC and consensus-based control is proposed in (58). The developed controllers allow the UAVs to fly in a steady formation under wind disturbances. However, the robustness of these controllers have not been field tested in the presence of significant communication perturbations. The formation control algorithms significantly rely on the communication for information synchronization and any disturbances to the synchronization degrades the performance of the formation controllers which is not taken care. The UAVs operate in air where they do not face the delays which can significantly hamper the controller performance, unlike the AUVs. Hence, these controllers cannot be directly used for AUV applications.

A team of AUVs are required to tightly coordinate for different kinds of applications as discussed above. A common method used for AUV formations are based on leader-follower strategy, (59, 60, 61, 62). In this method, the leader is given the reference trajectory and the follower tracks the reference trajectory without having a knowledge of the trajectory but by taking the leader position and the predetermined formation constraints. There are other methods that typically use a decentralized controller by considering the neighbors information and trajectory reference into account to feed the ormation controllers. Jia and Li, (63), developed a potential function and behavior rules to effectively control the AUV formation under uncertain environmental conditions with obstacle avoidance. Yang and Zhang, (64), use Hamilton-Jacobi theory and geometric reduction techniques for formation control. Yang and Gu, (65), develop a smooth feedback control law using Lyapunov direct method to enable a stable formation and a time varying smooth feedback control law using the integrator backstepping method is designed to collaboratively moor the follower AUV to its desired docking position and orientation with respect to the leader. Cui et al., (66), present a formation controller that needs to maintain a fixed topology. They use an adaptive sliding variable structure control to achieve the formation. In (67), the authors (Yan et al.) use a combination of backstepping and Lyapunov method to derive the path following algorithm for each AUV, and all AUVs share the path information to achieve the formation task. Shen et al., (68, 69), uses MPC to solve path following control problems of a AUVs.

Chapter 3

Model Predictive Control. Applications for Autonomous Underwater Vehicles

3.1 Introduction

As mentioned in the introduction, this thesis proposes an efficient approach to the optimized control of a single or multiple ARVs. Moreover, the approach will be illustrated in the context of AUVs chosen for two reasons: a) LSTS (http://lsts.fe.up.pt/) made available an infra-structure allowing the characterization of the key challenges as well as to assess the proposed framework, and b) constitutes a good representation of an instance in which control often amounts to the optimization of onboard resources in the presence of severe constraints and significant perturbations.

The need to synthesize optimal or near optimal control strategies in a context of very diverse performance functionals and constraints of various types leads us to consider optimal control problem based frameworks due to its huge versatility. Moreover, the need to deal with both small and wide perturbations implies the need of considering closed loop control schemes. Model Predictive Control (MPC), also known as Receding Horizon Control (RHC), brings together these two key aspects of the control problem.

We would like to point out that this scheme might be considered in two quite different perspectives: (a) optimization over a receding horizon, or (b) approximation of an optimization over a given time horizon (possibly infinite) by a sequence of much shorter time horizon optimization problems. The later has been widely considered in the literature and will be the one adopted here.

Thus, in this chapter, we start by outlining the general MPC scheme, and, then, we present a sample of some of previous work using MPC schemes to control AUVs and formation of AUVs. We emphasize that the literature on both these two general topics is huge. Thus, we will focus in two main approaches to MPC as well as in the overview of the key properties, notably, stability, robustness, and optimality. The issues addressed in this exposition will serve as a basis for the research underlying their transposition to the context of the novel Attainable Set MPC presented and discussed in Chapter 5.

3.2 Description of a representative MPC scheme

MPC is a control scheme in which the control action for the current time subinterval – the control horizon – is obtained, at each sampling instant, by solving on-line an optimal control problem over a certain large time horizon – the prediction horizon – with the state variable initialized at the current best estimate updated with the latest sampled value. Once the optimization yields an optimal control sequence, the control in the first control horizon of this sequence is applied to the plant. Then, once this time period elapses, the process is re-iterated. Let t_0 be the initial time, x_0 denote the associated initial state estimate, T is the prediction horizon for control optimization, and Δ is the control horizon. Thus, the MPC scheme is as follows:

1. Initialization. Let t_0 be the current time and x_0 denote the associated current state estimate, and set up the initial parameters or conditions specifying T, Δ, initial filter parameters (in case the sampled data requires filtering, initial control for the recursive control optimization procedure, cost functional weights, model parameter estimates, etc.

2. Sample the state variable at time t_0 and generate the associate state estimate, x_0.

3. Compute the optimal control strategy, u^*, in the prediction horizon, i.e., $[t_0, t_0 + T]$, by solving the optimal control problem (P).

4. Apply the obtained optimal control during the current control horizon, $[t_0, t_0{+}\Delta]$.

5. Slide time by Δ, i.e., $t_0 = t_0 + \Delta$, and adapt parameters and models as needed.

6. Goto step 2.

A number of variants to this scheme have been considered by enriching some of steps with additional processing capabilities:

- For the networked systems implementation, the data obtained in step 2. might consist of a composition of locally sampled data and data communicated from other vehicles or subsystems. For this class of systems, it might be of interest to replace data that failed to be transmitted by simulated data.

- Filtering the sampled state variable (say, by using a Kalman filter) might be required to produce a state estimate.

- For situations in which models are significantly uncertain or may vary over time, it might be of interest to use the sampled data to identify or refine the model parameters values.

- Likewise, if external perturbations or uncontrolled inputs acting on the vehicles/systems are sensed or otherwise estimated, they can be used to improve the models entering in the optimization procedure, as well as, to change the MPC parameters.

- Communication may introduce delays and data packets might fail to arrive with serious consequences in the controller performance. These issues can be addressed by either replacing true data by simulated data and/or adjusting the MPC parameters.

Let us consider now a typical general formulation of the optimal control problem (P). Consider a given fixed time interval $[t_i, t_f]$.

$$
\begin{aligned}
(P)\ \text{Minimize} \quad & g_0(x(t_f)) + \int_{t_i}^{t_f} f_0(t, x(t), u(t))dt \\
\text{subject to} \quad & \dot{x}(t) = f(t, x(t), u(t)) \quad \mathcal{L}\text{-a.e. on } [t_i, t_f] \\
& u(t) \in \Omega \quad \mathcal{L}\text{-a.e. on } [t_i, t_f] \\
& h(t, x(t)) \leq 0 \quad \forall t \in [t_i, t_f] \\
& l(t, x(t), u(t)) \leq 0 \quad \forall t \in [t_i, t_f] \\
& x(t_i) \in C_i \quad x(t_f) \in C_f,
\end{aligned}
$$

where $g_0 : I\!R^n \to I\!R$ is the endpoint cost functional, $f_0 : I\!R \times I\!R^n \times I\!R^m \to I\!R^n$ is the running cost integrand, $f : I\!R \times I\!R^n \times I\!R^m \to I\!R^n$, $h : I\!R \times I\!R^n \to I\!R^k$, and $l : I\!R \times I\!R^n \times I\!R^m \to I\!R^q$ represent, respectively, the vehicle dynamics, the state constraints, and the mixed constraints. $C_f \subset I\!R^n$ is the target set which, besides having a physical meaning, is often used in several approaches to ensure stability. Notice that the state variable starting point is a decision variable, and, thus, the minimization takes place over all the initial set $C_i \in I\!R^n$.

We point out to the following three general approaches to solve the optimal control problem, (70):

(i) Recursive procedure to solve a $2n$ boundary value problem based on the Pontryagin Maximum Principle (PMP) that yields an open loop optimal control strategy. Since any feasible control process satisfying the PMP conditions is only an extremal, i.e., a candidate to solution to the optimal control problem, this approach configures an elimination procedure. That is, the PMP discards from the set of optimal control solution candidates all the feasible control processes that do not satisfy its conditions. Usually, an additional step is required in order to select the optimal control process from the set of extremals.

(ii) In the case in which the Value Function, i.e., $V(t, x) := \min_u \{J(u)_{|[t,t_f]} : x(t) = x\}$ where $J(u)$ is the cost functional as a function of the control, associated with the problem satisfies the principle of optimality, then the Value Function can be obtained as a solution (in an appropriate sense which depends on the regularity of the problem and the activity of the constraints along the optimal control process) the Hamilton-Jacobi-Bellman equation associated with the control problem. Thus, in this case, the solution is obtained in a closed loop functional, that is an optimal control trajectory is given for each point (t, x) travelled by the optimal trajectory. Dynamic programming algorithm and approximating linear or nonlinear mathematical programming schemes have been used to obtain numeric approximations to the solution of (P); and

(iii) Algorithms for solving finite dimensional nonlinear optimization problems obtained by considering discrete approximations to the, typically nonlinear, programming problems in infinite dimensional spaces in which problem (P) can be formulated.

While the first two approaches are designated of indirect methods because they use optimality conditions in an intrinsic way, the third one falls in the set of the so-called direct methods.

Usually, the computational complexity of solving problem (P) is very high and a number of approaches have been considered to address it. Some of these will be discussed later in this section.

The MPC scheme exhibits some welcome features, of which we consider some next:

- It replaces solving an long (possibly infinite) time horizon optimal control problem by the computation of the solution to a sequence of open loop optimal control problems with receding shorter time horizons and the state variable initialized with the sampled value. This enables the distribution of the computational effort over time and, thus, makes it particularly amenable to satisfy real-time requirements. Moreover, since the reference trajectory variations from one iteration to another are, in general, small, it is possible to ensure a good initialization of the iterative optimal control algorithms, and, thus, the whole process becomes very efficient.

- It enables the implementation of state feedback control strategies since each one of the shorter horizon optimal control problems is solved with the state variable initialized with the sampled state. Moreover, in opposition to other controllers for which a feedback control policy is determined off-line (by solving, for example, an

Hamilton-Jacobi-Bellman differential equation or its discrete version), the MPC solves an optimal control problem on-line for the current state of the plant.

- The fact that it involves a sequence of open loop optimal control problems, makes this scheme extremely suitable to deal with versatile state and or control constraints. This feature is important to model complex problems like, for example control formations.

In the next sections we will address key issues pertinent to the MPC scheme. These consist essentially in: stability, robustness, computational tractability, and sub-optimality. There is an extremely wide body of literature for the standard MPC scheme and so we will focus in a small but representative sample.

3.3 Stability, robustness, uncertainty, tractability and optimality

Stability

Stability properties are of major importance as, otherwise, any perturbation may drive the system to undesirable states. When naively designed, the stability of the MPC scheme is not guaranteed. Stability of linear system is well studied today but in non-linear systems there has been an intense research over the last 30 years and a very wide range of results are available today. These results are different in nature and depend strongly not only on the adopted notion of stability, but also on the chosen specific set up of the MPC scheme as well as in the approach used to establish it (1, 71, 71, 72, 72, 73, 73, 74, 75).

While some handle the continuum time control system, others focus the discrete time variety. There are several approaches to show that, under appropriate set of conditions, diverse MPC schemes generate sequences of state feedback control laws, say $u = k(x)$, so that the associated sequence of trajectories converge to an equilibrium point. Early on, results were derived for linear systems which are now, well established. Then, a lot of research effort has been put in extending these results to the nonlinear context.

Naturally, capitalizing on these, approaches linearizing the dynamics around equilibrium points, and, thus, transforming the given nonlinear system into a linear piecewise one have been used. Terminal-cost based approach which includes setting up the MPC so that the objective function of the optimal control problem includes a term dependent on the terminal state. For linear systems this term can be constructed with the solution to an algebraic Riccati equation. Nonlinear systems also use a terminal cost in the objective function to achieve stability. The idea is to choose a the terminal cost such that it exceeds the running cost until the infinity. Global Control Lyapunov Functions (CLF) are also quite popular. If a global CLF can be found, or it can be shown that the Value Function, computed by Dynamic Programming techniques or by solving the Hamilton-Jacobi equation, satisfies the properties of a CLF, then a stabilizing feedback can be found without having to solve the on-line optimization required by the MPC strategy. Other approaches consist in showing that as the iterations of the MPC scheme progress the Value Function decreases and, at the same time the level sets of the Value Function are invariant, that is the generated strategy remains within a set shrinking to a sufficiently small neighborhood of the equilibrium.

Regardless of the approach, a number of formulations involving either terminal state constraint set, C, or terminal cost f_0, or both, have been considered.

These can be organized into two major approaches:

a) Direct method using the fixed horizon Value Function as a Lyapunov function; and

b) Indirect approach employing the monotonicity property of a sequence of Value Functions.

Here, we have to content ourselves with providing a flavour of a landmark result of indirect type derived in (76)

We will start with by considering the key results in (76) in which an indirect approach is considered to prove stability under some reasonable assumptions. In this work, the dynamic system operates in $[0, \infty)$, perturbations are not considered explicitly, the state variable at the initial time is given, no state or mixed constraints are considered and the nonlinear dynamics are time-invariant. The following standing assumptions on f are considered

- $u(t) \in \Omega \subset I\!R^m$ where Ω is a compact, convex set with 0 in its interior.

- $f : I\!R^n \times I\!R^m \to I\!R^n$ is twice continuously differentiable and $f(0,0) = 0$. Thus, when $u = 0$, $x = 0 \in I\!R^n$ is an equilibrium of the system.

- For any feasible piecewise continuous control $u : [0, \infty) \to \Omega$, the dynamics has a unique solution for any initial state $x(0) = x_0 \in I\!R^n$.

The optimal control problem considered in setting up the MPC scheme is as follows:

$$\begin{aligned} (P_1)\text{Minimize} \quad & J(x(t), \bar{u}) = \|\bar{x}(t+T)\|_P^2 + \int_t^{t+T} [\|\bar{x}(s)\|_Q^2 + \bar{u}(s)\|_R^2] ds \\ \text{subject to} \quad & \dot{\bar{x}} = f(\bar{x}, \bar{u}), \quad \bar{x}(t) = x(t) \\ & \bar{u}(s) \in \Omega \quad \forall s \in [t, t+T] \\ & \bar{x}(t+T) \in \Xi. \end{aligned}$$

Here, T is a certain finite prediction horizon, $\bar{u}$ is the control strategy obtained by solving (P_1), $\bar{x}$ is the trajectory associated with $\bar{u}$ with $\bar{x}(t) = x(t)$, P, Q, and R are certain positive definite weighting matrices. The terminal constraint $\bar{x}(t+T) \in \Xi$ ensures that the state variable at $t+T$ is in some neighborhood of the origin which is chosen so that it is invariant for the nonlinear system controlled by some local linear state feedback $\bar{u} = K\bar{x}$. The terminal quadratic $\|\bar{x}(t+T)\|_P^2$ is an upper bound to the infinite horizon cost starting from Ξ and controlled by the linear feedback $\bar{u} = K\bar{x}$, i.e.,

$$\|\bar{x}(t+T)\|_P^2 \geq \int_{t+T}^{\infty} [\|\bar{x}(s)\|_Q^2 + \bar{u}(s)\|_R^2] ds,$$

being P and Ξ chosen a priori so that, together with the linear feedback control law and other parameters, X_i is invariant when the input constraints are satisfied by the local linear state feedback, i.e., $\bar{u} = K\bar{x}$ $\forall \bar{x}$ such that $\bar{x}(t+T) \in \Xi$.

Thus the main result here is as follows.

Theorem 3.3.1 *Let the above standing assumptions hold, the Jacobian linearization of the given nonlinear system be stabilizable, and the open-loop optimal control problem P_1 be feasible for $t = 0$. The set X_0 of all initial values x_0 for which the last condition holds is called the attraction region for the closed-loop system. Then, in the absence of disturbances, for a sufficiently small sampling time Δ, the closed-loop system obtained by applying the MPC scheme is asymptotically stable.*

Clearly, the idea behind proof steams from the setup adopted in (76) which is to guarantee infinite horizon stability of the closed-loop system by determining the control function for a certain finite prediction horizon.

The linear state feedback is only used to a priori determine a terminal penalty matrix P and a terminal region Ξ. The desired invariance property of Ξ is obtained by assuming the stabilizability of the origin for the linear dynamic system obtaining by the Jacobian linearization the given dynamic system around $(0,0)$, and a procedure is offered to determine Ξ, P, and K.

The proof proceeds by showing the existence of a feasible control to (P_1) for all $t \geq 0$, for a sufficiently small sampling time Δ. Also, in order to obtain the asymptotic stability of the closed-loop system, it is required to show that the optimal Value Function is non-increasing, i.e., $\forall t \geq 0$, and $\forall s \in (t, t+\Delta]$, the optimum cost functional (denoted by J^*) satisfies

$$V(x(s)) \leq V(x(t)) - \int_t^s [\|\bar{x}(\sigma)\|_Q^2 + \bar{u}(\sigma)\|_R^2] d\sigma,$$

where $V(x(\tau)) := J^*(x(\tau))_{|[\tau,\tau+T]}$.

In order to conclude the proof it is enough to show that (i) $V(0) = 0$ and $V(x) > 0$ $\forall x \in X$ such that $x \neq 0$; (ii) $V(x)$ is continuous at 0; and (iii) along any close loop trajectory starting at $x_o \in X_0$ we have for any t_1 and t_2 such that $0 \leq t_1 < t_2 < \infty$, that

$$V(x(t_2)) - V(x(t_1)) - \int_{t_1}^{t_2} \|x(t)\|_Q^2 dt.$$

This is so, since from these properties of V, it follows from the fact that V is non-negative and bounded, x is uniform continuous on $[0, \infty)$, that, by applying Barbalat's Lemma, that $\|x(t)\| \to 0$ as $t \to \infty$, that is 0 is asymptotically stable, and, thus $X_\beta = \{x \in X : V(x) \leq \beta\}$ is a region of attraction. A simple contradiction arguments shows that (i) any trajectory starting in X enters X_β in finite time, and (ii) X is an invariant set to the MPC closed loop system.

<u>Robustness</u>

The issue of Robustness is very important as it is required in most of practical applications. Robustness concerns the ability of the system in preserving a certain property - e.g., stability or performance - in the presence of uncertainties. For stability, this can be checked by concluding that the Lyapunov function for the nominal closed-loop system keeps the descent property for sufficiently small disturbances. While this is not very difficult to show for unconstrained problems, the consideration of constraints on states and controls raises substantial challenges as it is required to ensure that the constraints remain satisfied. Inherent robustness, min-max open loop control, and feedback control are the general contexts considered to investigate the robustness of MPC schemes. While the first one concerns the robustness of closed loop systems, designed using the nominal system, the second attempts to achieve robustness in the context of a conventional MPC scheme considers all possible realizations of the uncertainty (min-max open-loop), and the third approach addresses this by introducing feedback in the min-max optimal control problem solved on-line.

Tracking, output feedback, adaptive model predictive control, optimization algorithms are some of other miscellaneous contexts in which stability and robustness have been addressed.

For a much more detailed overview, consult (73, 77).

Uncertainty

Uncertainty is extremely pervasive, and the more so in the underwater milieux. This is due not only to the complexity of the underwater environment but also to the fact that hydrodynamic phenomena are of a distributed character whose intrinsic complexity is typically circumvented by considering approximate concentrated parameters models. Two main approaches have been considered to handle uncertainty:

- Replace the reference trajectory by a tube. A scheme of feedback MPC that overcomes disadvantages of the conventional scheme with a manageable computational complexity consists in solving an on-line optimal control problem to obtain a "tube" and the associated piecewise affine control law that maintains the controlled trajectories in the tube despite uncertainty (see (71, 72)). Some of the key features of the scheme proposed in this paper are the linearity of the computational complexity in horizon length and the asymptotic stability of the controlled system.

- Couple receding horizon estimation and control. The problem of output feedback MPC of discrete time systems in the presence of additive but bounded state and output disturbances is considered in (73). Here, the scheme involves a stable state estimator and a tube based, robustly stabilizing model predictive controller. This scheme allows to extend earlier results to time varying estimators. By requiring the online solution of a standard quadratic program, the proposed robust output feedback controller ensures that a specified invariant set robustly exponentially stable.

Computational tractability

Although, in the early stages, MPC schemes became popular with large systems with relatively slow dynamics, the recent rapid progress of computation and communications technologies made it possible to consider a whole new and wide range of applications involving control systems with much faster dynamics. As a consequence, a lot of work concerned computational issues pushing the limits in handling problems with very high computational complexity subject to very hard real-time constraints. As examples, one may consider power electronics, energy management, machinery automation, automotive applications, etc. In this context, the need to address issues inherent to handling problems with very high computational complexity subject to very hard real-time constraints emerged.

The need to ensure the computationally tractability in the optimization of discrete time linear hybrid systems, the modeling framework Mixed Logical Dynamical (MLD) models for PieceWise Affine (PWA) dynamic systems has been developed, see (78). Polyhedral PWA systems are defined by partitioning the input-state space into polyhedra and associating with each polyhedron an affine state-update and output function and it can be regarded as a computationally efficient way of dealing with nonlinear systems. Besides being a well posed framework to bridge the continuum time driven dynamics and the logical world, the MLD modeling framework is also amenable for the tool HYSDEL, (http://cse.lab.imtlucca.it/∼bemporad/hybrid/toolbox/). This tool provides a high level, intuitive textual interface for modeling a class of hybrid systems described by interconnections of linear dynamic systems, automata, if-then-else and propositional logic rules, known as Discrete Hybrid Automata (DHA).

An application in the context of power electronics - with very fast dynamics - is considered in (79). In order to accommodate the computational burden with very fast dynamics, the following approach was adopted:

i) Pre-solve off-line optimization problems for the whole state-space using multi-parametric programming - leading to a PWA controller stored in a look up table.

ii) Use dedicated solution algorithms.

In (80), a MPC algorithm is proposed for the robust control of continuous-time systems. Discontinuous feedback strategies are produced as solutions of min-max problems. The use of bang-bang feedbacks described by a small number of parameters reduces considerably the computational burden associated with solving a differential game. Affine controls of the bang-bang type are pre-computed off-line and then selected on-line. The applicability of the proposed algorithm is tested to control a unicycle mobile robot.

A similar idea is used in (79) to apply a MPC scheme to optimize the behavior of a power electronics system with extremely fast dynamics. In (81), a precious advantage is taken from available efficient linear quadratic solvers to address the real time constraints associated with the control of formations of aerial vehicles.

Optimality

Clearly, if the dynamic system is linear, the cost quadratic and the constraint sets have a finite and affine representation, the optimal open-loop control problem reduces to a quadratic programme for which efficient software packages yielding a global solution to the optimal open-loop control problem exist. In this context, it is not hard to see that the model predictive controllers yield near global optimal solutions.

However, in the case of nonlinear dynamic systems, usually, the open-loop optimal control problem is non-convex. Nonlinear programming algorithms usually yield only local solutions and it is reasonable to investigate whether the needed MPC properties if global solutions to the optimal open-loop control problem are not obtained. This difficulty is overcome if the approach required to prove the stability of the MPC scheme only requires feasible solutions to the constrained optimal control problem since the verification of the feasibility is computationally simple. Such approaches have been considered in (82, 83) for continuum time problems, and (84) extend this strategy for discrete-time systems.

Of course to achieve optimality is always preferred in many instances for which performance of the system is important. However, when achieving optimality is not viable, several techniques, such as, settle for the current suboptimal control process when the real-time constraint does not allow further progress of the optimization procedures, or use a simpler version of the optimal control problem for which the complexity is compatible with the real-time constraints.

3.4 Approaches to MPC based AUV formation control

There is an extremely vast body of literature on MPC that we can not hope to include in this overview. See, for example, (72). We will focus on the key results that are pertinent to our approach and focus on the class of systems addressed in this work, the coordinated control of formations of vehicles.

The versatility exhibited by optimal control problems has been exploited in order to formulate and solve problems of controlling formation of vehicles. These typically have a substantially complex structure and may be addressed by using MPC schemes in either a decentralized or a centralized context which may involve two stages: the planning phase - solved off-line to provide a reference trajectory for the formation of vehicles -, and the execution phase - solved on-line with the help of locally formulated control problems. Let us overview a selected sample of some of these approaches.

In (85), it is proposed a "control architecture" for networked systems. Given the fact that information of both local and global nature is required in either "Leader-Follower" or "Shared cooperation burden", the control objective for each vehicle encompasses two types of components:

(i) a local one, of a relative nature, obtained by local sensing or communication with neighbors; and

(ii) a global one, typically obtained by communication.

Thus, in this approach, the control law for each one of the systems is arranged in an additive way in two components: feedback and feedforward. A receding horizon control strategy is obtained by considering a finite horizon, integral quadratic cost function reflecting the local objective as well as the formation constraints with neighbors defined by a graph.

In (86), the problem of cooperative control of a team of distributed agents with decoupled nonlinear discrete-time dynamics, which operate in a common environment and exchange-delayed information between them is considered. Each agent is assumed to evolve in discrete-time, based on locally computed control laws, which are computed by exchanging delayed state information with a subset of neighboring agents. The cooperative control problem is formulated in a receding-horizon framework, where the control laws depend on the local state variables (feedback action) and on delayed information gathered from cooperating neighboring agents (feedforward action). A rigorous stability analysis exploiting the input-to-state stability properties of the receding-horizon local control laws is carried out. The stability of the team of agents is then proved by utilizing small-gain theorem results.

A decentralized scheme for the coordinated control of formations of autonomous vehicles is presented in (81) that builds on the work reported in (87). A high level receding horizon control and coordination strategy is obtained for each vehicle by solving a linear quadratic optimization problem featuring control saturation constraints, linear dynamics constraints, and formation constraints with neighboring vehicles defined by a graph. An appropriate graph structure describes the underlying communication topology between the vehicles. On each vehicle, information about neighbors is used to predict their behavior and plan conflict-free trajectories that maintain coordination and achieve the team objectives. When feasibility of the decentralized control is lost collision avoidance is ensured by invoking emergency maneuvers that are computed via invariant set theory. A stabilization analysis is also discussed in (87).

Information exchange strategies that improve formation stability and performance and, at the same time, are robust to changes in the communication topology are considered in (88) to address the problem of cooperative control of vehicle formations. The sensed and communicated information flow is modeled by a graph whose topology may have implications in control stability. By exploiting the interplay betweens communications and control, necessary and sufficient conditions for the stability of an interconnected system of identical vehicles. Stated in terms of the Popov criterium for networked control systems, these conditions involve the eigenvalues of the graph Laplacian and reveal how to shape the information flow in order to ensure stability and achieve high performance.

Robust stability results of the Popov type for networked systems are presented in (89). By using integral quadratic constraints (IQCs), the interconnection structure is exploited to decompose the analysis of the overall system into lower dimensional sub-problems leading to a significant reduction of the computational complexity. In a heterogeneous network, where the local dynamics are different but characterized by the same IQC, the authors show that the analysis problem decomposes when the interconnection matrix is normal. Then, a class of Popov criteria for networks with symmetric interconnection matrices is identified and the criterion is obtained by using a IQC characterizing the symmetric interconnection matrix has real eigenvalues within a certain range.

Of particular interest for the decentralized control of formations is the problem of string stability that has been addressed by a number of authors, (90, 91, 92). String stability provides a measure on how the position errors propagate from one vehicle to another in a formation in which each vehicle regulates its motion only relatively to its neighbors. In (92), it is provided a characterization of the impact of communications delays in the string stability of a highway platoon. The analysis was carried out in the context of linear longitudinal models and a solution to counter this sensitivity was provided. Autonomous aerial vehicle formations is considered in (91), for which linear and nonlinear formation performance simulation analysis are carried out. Recommendations for control design are provided by applying string (un)stability results in the context of several classes of perturbations. Two interesting string stability results for an infinite string of cascaded identical linear systems are provided in (90). While one provides sufficient conditions for string stability in terms of the eigenvalues of the state evolution matrix of a system obtained by a discrete Fourier transform applied to the original system, the other concerns the equivalence of the string stability condition for both the state-space and frequency-domain representations.

The consensus problems for networks of dynamic agents with fixed and switching topologies are discussed in (93) in which three cases:

i) directed networks with fixed topology,

ii) directed networks with switching topology, and

iii) undirected networks with communication time delays and fixed topology.

Two consensus protocols for networks with and without time-delays are considered and a convergence analysis establishes a connection between the algebraic connectivity of the network and the performance of a linear consensus protocol.

In (94), an approach to design controllers for a team of agents that accomplish consensus for agents' output in both leaderless (LL) and modified leader-follower (MLF) architectures is presented. Towards this end, a semi-decentralized optimal control strategy is designed based on minimization of individual cost functions over a finite horizon using local information. Interactions among agents due to information flows are represented through the control channels in characterization of the dynamical model of each agent. It is shown that minimization of the proposed cost functions results in a modified consensus algorithm for LL and MLF architectures.

The work (95) discusses the design of control strategies for multivariable plants where the controller, sensors and actuators are connected via a digital communications channel with data-rate constraints. In order to minimize the bandwidth utilization, constraints on communications are imposed to restrict all transmitted data to belong to a finite set and to permit only one plant to be addressed at a time. The implementation issues and moving horizon techniques to deal with both control and measurement quantization issues are emphasized and the methodology is illustrated by simulations as well as a laboratory-based pilot-scale study.

In (96), a two-layer scheme to control a set of vehicles moving in a formation is proposed. The first layer consists of a MPC trajectory controller. It is a nonlinear since, in general, most vehicles are nonholonomic and may even require a discontinuous feedback controls in order to be stabilized. It computes centrally a bang-bang control law so that only a small set of parameters has to be transmitted to each vehicle at each iteration. The second layer consists in the formation controller. Since it aims to compensate for small changes around a nominal trajectory maintaining the relative positions between vehicles, this second layer can be adequately carried out by a linear model predictive controller accommodating input constraints and state constraints. This has the advantage of simplifying the control laws for each one of the vehicles. These are simple piecewise affine feedback control laws that can be pre-computed off-line and implemented in a distributed way in each vehicle.

The problem of unreliable communication channel between the MPC controller output and the actuator input, has been addressed in, among others, (97) where a mechanism for compensation of packet dropouts has been incorporated in the MPC scheme for discrete time problems. The basic idea consists in extending the applied control subinterval until the next successful communication event happens and, in the meantime, use the best available control estimate, namely the one that has already been computed for the longer time interval. This reference also includes some stability and sub-optimality analysis under an asymptotic controllability assumption. In order to show stability, the authors prove that, under the considered assumptions, the Value Function associated with the optimal control problem exhibits properties of a Lyapunov function.

Chapter 4

An Implementation of a Conventional MPC for AUV Formations

4.1 Introduction

In this chapter, we develop and implement an MPC based scheme for the decentralized control of a leader-follower formation with rigid relative positions among the vehicles that was developed in the context of the FP7 project "Control for Coordination of Distributed Systems" – FP7-ICT-223844 – with the support of the Laboratory for Underwater Systems and Technologies (LSTS) of FEUP with vehicles and software.

The control problem consists in tracking a given trajectory while keeping a pre-specified formation which is defined by the distance between any two vehicles and the angles the vector defined by their positions form in a given reference frame. The approach consists in making available to each vehicle its own reference trajectory, and the controller of each vehicle will have to correct the very likely emerging errors in the vehicles relative positions in the course of the mission.

In the MPC literature, the specific issues associated with AUVs, typically the scarcity of on-board resources, such as power, computation, and data from communications, are usually weakly addressed. Due to this reason, we cannot use them to the AUV formation control under real world considerations which is a key contribution of this thesis. To consider all the practical issues and provide a feasible controller, we use

MPC. MPC has been used widely for formation control as well as for other applications. However, the developments reported in this chapter target practical implementation issues, notably, low data rate and delays inherent to acoustic communications as well as the other strict on-board resources constraints mentioned above.

In our scenario, each vehicle runs an MPC algorithm that, by taking into account its own and its neighbors navigation data, generates a control strategy that balances the minimization of the quadratic error to the reference trajectory and that of the deviation from the pre-specified formation pattern with the minimization of the employed control effort over a given time interval (the prediction horizon). Control and state constraints are also considered in order to reflect control saturations as well as to avoid the collision with obstacles. Here, we are clearly taking advantage of the enormous flexibility of the optimal control paradigm.

The obtained control is applied for a short time interval (the control horizon), after which the state is sampled and information is exchanged among the pertinent neighboring vehicles via an acoustic communication channel. Then, the cycle is restarted with the new optimization carried out over a shifted prediction horizon with the most recent data (either sampled or estimated).

The decentralized nature of this problem - due to the partiality of the information available to each AUV - calls for a level of communication among vehicles and of computation in each vehicle that strongly conflict with the available onboard resources. Thus, two main issues may arise in the networked MPC scheme:

- One concerns underwater acoustic communication which may exhibit delays (due to the low sound propagation velocity) and packet loss. Still, this information enables to close the control loop, and thus, to increase the robustness of the control strategy. The communication delay on the sampled data sent in by other vehicles is partially compensated by generating a model prediction using the most recent data. To tackle packet dropouts, the "redundancy" of the MPC scheme is exploited by adjusting its parameters (e.g., short control application time interval is extended), being the computation of the new optimal controls triggered by the next successful data communication event. Until then, previously computed optimal controls are applied.

- Another issue of importance concerns the computational complexity which, while taking into account the strict limitations of the AUV onboard resources, also has to meet hard real-time constraint requirements. A reformulation of the MPC in the context of Attainable Sets, will be addressed on Chapter 5.

A substantial amount of research work has been done on the control, and a few on MPC schemes, of formations of autonomous vehicles, (81, 85, 86, 87, 88, 91, 92, 93, 94, 95, 96, 97, 98, 99). However, to the best of our knowledge, there are no satisfactory developments that fulfill the requirements that we encountered in our application.

4.2 Optimal control problem formulation

The general problem concerns the decentralized control of a set of vehicles that should move while satisfying certain given formation constraints – which may either remain time invariant or evolve dynamically – in order to accomplish the specified mission objectives according to some given requirements.

Without a great loss of generality, in this work, we will focus in the special case in which each one of the AUVs track a given trajectory and, at the same time, the set of the vehicles has to maintain a given formation pattern. Additionally, the vehicles should be able to avoid collision with unanticipated obstacles and to switch between pre-specified different formation patterns. This means that the control system of each vehicle will need its own navigation data and the one communicated by other vehicles in order to define an actuation that accomplishes its objectives and, at the same time, satisfies hard constraints.

The design of such a control system is by no means an easy problem since it exhibits a wide variety of extremely challenging features. These steam from the strict limitation of on-board resources, the "opacity" and "hostility" of the environment, and the AUV motion modeling complexity.

In fact, resources onboard the vehicles – space and power – are at a premium. Operational considerations and cost effectiveness of the overall system and its operation dictate a bound on the size of the AUVs, which, in turn, limits the amount of hardware (batteries, actuators, sensors, signal and power electronics, computational systems, communication devices, etc.) required for the functioning of its subsystems. Given the currently available technologies, this means that most power hungry activities –

actuation, sensing, communication and computation – have to be carefully balanced in order to accomplish the mission with the specified requirements and, at the same time, optimize the performance of the whole system.

The underwater environment is difficult. Besides the complexity of the evolving multiple phenomena, there are two fundamental aspects: (i) the propagation of signals is such that transmission of information is currently viable only at data rates much lower than those in the atmosphere; and (ii) the hydrodynamic effects of underwater phenomena are powerful elements difficult to model and to take into account in such a way to ensure that AUVs achieve the desired goals.

A complete six-dof model of an AUV is a complex task to achieve due to model couplings. However, there exists simplified decoupled models that are non-interacting. These simplified models together with the perturbations significantly affects the AUV behavior. This causes uncertainty in predicting its behavior. Moreover, underwater acoustic communications are usually not only unreliable (packet dropouts) but also might introduce non-negligible delays due to the relatively low velocity of the sound propagation in the water. This calls for mechanisms built in the control framework to increase the robustness of the designed control system.

Sensory data and information for navigation and motion control are either costly or of poor quality: While the covariance associated with proprioceptive data typically increases rapidly over time, the exteroceptive data – e.g., GPS or LBL (triangulation of distances to acoustic transponders with known positions) – is costly since it requires surfacing in the former and the interrogation of acoustic transponders in the later. Furthermore, the LBL system limits AUV operations to a finite area. Since the AUV motion in a formation requires the control loop to be closed with data from its neighboring vehicles, communication, typically acoustic, has to be established and this is also very expensive resourcefully wise.

The above considerations makes the case for a control framework in which the synthesized control fulfills the following key requirements: (i) state feedback form; (ii) decentralized nature; (iii) optimizing on-board resources; (iv) together with its state trajectory, satisfying all the constraints; and (v) ensuring additional pre-specified behaviorial properties such as stability, robustness, sub-optimality, etc.

Let us consider a formation of n_v AUVs tracking a given trajectory η^i_{ref} where, for each vehicle i, η^i is the position, orientation, linear and angular velocities, and τ^i the vehicle restoring forces and moments. We are interested in control strategies that, for AUV_i, $i = 1, \ldots, n_v$, minimize, over a given time interval T, a cost functional with two terms, one that penalizes the trajectory tracking error forcing vehicles to follow the desired path, η^i_{ref}, and another that penalizes the control effort, thus saving the limited power on board of vehicles, i.e.,

$$\int_t^{t+T} \left[(\eta^i(s) - \eta^i_{ref}(s))^T Q(\eta^i(s) - \eta^i_{ref}(s)) + {\tau^i}^T(s) R \tau^i(s)\right] ds, \tag{4.1}$$

and, at the same time, satisfies the following:

(i) Kinematic and dynamic equations constraints;

(ii) Endpoint state constraints, $\eta^i(t+T) \in C_{t+T}$;

(iii) Control constraints, $\tau^i(s) \in \mathbb{U}^i$;

(iv) State constraints, $\eta^i(s) \in \mathbb{S}^i$;

(v) Communication constraints $g^c_{i,j}(\eta^i(s), \eta^j(s)) \in C^c_{i,j}$, $\forall j \in \mathcal{G}^c(i)$; and

(vi) Formation constraints $g^f_{i,j}(\eta^i(s), \eta^j(s)) \in C^f_{i,j}$, $\forall j \in \mathcal{G}^f(i)$.

The vehicle's kinematic and dynamic equations in (i) will be discussed later in Section 4.3. For the sake of stability, the endpoint state constraints are bound in the set C.

The control constraints (iii) include, for example, actuators saturations, and the constraints in (iv) are imposed to keep each vehicle in a specified set in order to satisfy safety requirements. For example, to avoid collision with – known a priori or detected on the fly – obstacles, or to prevent some variables to take on values that may damage components.

The satisfaction of acoustic communication constraints (v) ensure that the motion of the vehicles is such that the required connectivity among the AUVs is preserved. The fact that the closer the vehicles are, the lower the power consumption and packets loss, makes a strong case for each AUV to communicate with its neighbors and, hence, for decentralized control structure. The communications structure may be described by triple $(g^c, C^c, \mathcal{G}^c)$ where $g^c : \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}^M$, $C^c \in \mathbf{R}^M$ (where $M \leq n(n_v - 1)n_v$ being n is the dimension of the relevant state space component of each vehicle), and

$\mathcal{G}^c$ is a graph whose i^{th} component defines the vehicles communicating with the i^{th} vehicle. We point out that the communication graph is, in general, quite different from the formation or control graphs that we will introduce next. For example, a vehicle might simply provide a communications relay service without supplying any formation specific data for motion control.

In what concerns the design of a control structure, it should be pointed out that redundancy in the communications connectivity might be necessary to achieve the required degree of robustness but we will not consider this at this time, but focus only on: (i) non-negligible delays due to relatively low velocity of sound propagation, and (ii) packets loss.

Finally, formation constraints (vi) specify relations between data (typically, relative positions) of AUVs which have to be maintained with the help of appropriate control activity. These relative positions are specified in order to ensure the desired task requirements (e.g., data gathering) undertaken by the AUVs formation. The formation structure may be described by triple $(g^f, C^f, \mathcal{G}^f)$ where $g^f : \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}^M$, $C^f \in \mathbf{R}^M$ (where $M \leq n(n_v - 1)n_v$ being n is the dimension of the state space component of interest of each vehicle), and $\mathcal{G}^f$ is a graph whose i^{th} component defines the vehicles with a formation relation with the i^{th} vehicle.

We observe that the specification of the control structure of a formation of vehicles also encompasses the distribution among the vehicles of the burden of coordination in order to sustain the formation. In the leader-follower option, there is the advantage of easier stabilization (just the leader, and each one of the vehicles in an isolated fashion) but also the disadvantage of poor reliability due to the total reliance on the leader and poor disturbance rejection properties. These drawbacks do not appear to such a great extent in the option with a more evenly distributed burden of coordination but stabilization becomes more difficult and communication and computation efforts become more intense. In the development of this work, these options are kept open.

Complexity issues on one hand, and issues related to ensure feasibility of the optimization procedure (possibly at the price of allowing some graceful degradation of the specifications) on the other hand, motivate an alternative formulation of the optimization problem in which formation constraints are eliminated and an additional term penalizing the violation of the state constraints is added to the cost functional.

An example, for the case with the distance between vehicles i and j (with $(i,j) \in \mathcal{G}^f$) is given by d^{ij}, the term to add to the cost functional (4.1) would be:

$$\int_t^{t+T} (\eta^i(s) - \eta^j(s) - d^{ij})^T L^{ij} (\eta^i(s) - \eta^i(s) - d^{ij}) ds.$$

One of the interesting challenges posed by the problem of controlling formations concern stability. Interesting results for linear systems have been derived, for example in (88), where a formalism in terms of graphs specifying the information flow for control has been adopted for the analysis of the closed loop control system. The derived stability conditions are expressed in terms of the eigenvalues of the Laplacian representing the communication graph. Other approaches (see for example, (97)) consist in showing that the associated Value Function exhibit properties of the type of a Lyapunov function.

Clearly from the above, the control system of each AUV will have to generate feedback control that will close the loop not only on its state but also on the state of some of other AUVs as specified by the formation pattern. This configures a networked optimal receding horizon control (or networked MPC) since: (a) the state variable of each one of the vehicles has to be sampled from time to time (at the end of the control horizon, i.e., the control application interval), and this data has to be exchanged among some of them as specified by the formation control requirements; (b) the above stated Optimal Control Problem (OCP) is solved with the initial state data generated in (a) and over a long time horizon starting at the sampling time (the prediction or optimization horizon).

The higher the sampling frequency, the better the control system is able to deal with the uncertainty. However, there are very hard obstacles that make it difficult to improve the performance of such control systems, from which we single out the following:

- Packet dropouts, and communication delays due to the propagation speed of sound in the water, the associated onboard computation time, and sensor response features which contribute to the decrease of the control performance and robustness, and even lead to instability.

- Computational complexity required by models, sensor data processing, and the optimization based control synthesis which strongly contrasts with of the limited onboard processing capabilities.

To tackle these challenges, this work encompasses research on novel MPC schemes that will be addressed in this chapter and, in turn, is built on the state-of-the-art developments described in the previous section.

4.3 Modeling

Modeling AUVs' motion is difficult. Because of the hydrodynamic effects, AUVs are distributed parameter systems and, thus, represented by extremely complex (relatively to the available onboard computational power) models. This calls for the consideration of a concentrated parameter approximating model, e.g., (100), such as

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \tag{4.2}$$

$$\dot{\eta} = J(\eta)\nu \tag{4.3}$$

where ν, and η are, respectively, the linear and angular velocities in the vehicle body fixed frame, and the position and orientation in the inertial frame, M, $C(\nu)$, and $D(\nu)$ are, respectively, the inertia and added mass matrices of the vehicle, the Coriolis and centripetal matrix, and the damping matrix, $g(\eta)$, and τ are, respectively, the restoring forces and moments, and the body-fixed forces from the actuators, and $J(\eta)$ is the transformation matrix relating both reference frames (100, 101, 102, 103, 104, 105).

Unfortunately, AUV model identification in this general context is a very difficult and expensive process due mainly to the large number of rigid-body and hydrodynamic parameters and the complexity of the required experimental setups. For this reason, we use the decoupled model described next with parameter values based on results in (106) and on our own field experiments. This has been experimentally shown to suffice to characterize many specific classes of AUV motions with a reasonable accuracy. We consider the modes of operation Surge, Yaw, Pitch, and Heave, and we obtained the respective models presented in Table 4.1.

The details of the modeling approach are discussed in the appendix C. Here we also describe very simple procedures to estimate the coefficients of the simpler models of each motion mode which encompass not only the identification algorithms but also the procedure to collect data.

Table 4.1: AUV simplified model

Motion mode	Model
Surge	$\frac{X_{u\|u\|}u\|u\|+F_{prop}}{(m-X_{\dot{u}})}$
Heave	$\frac{mU_0q+Z_qq+Z_ww+Z_\delta\delta_s}{(m-Z_{\dot{w}})}$
Pitch	$\frac{-z_GW\theta+M_qq+M_ww+M_\delta\delta_s}{(I_{yy}-M_{\dot{q}})}$
Yaw	$\frac{N_rr+N_vv+N_\delta\delta_r}{(I_{zz}-N_{\dot{r}})}$

4.4 From optimal control to linear quadratic programming

In this section, we describe the implementation of a simulation environment for the decentralized version of a discrete time MPC system to control a formation of AUVs. The overall structure of the MPC simulation environment can be viewed in figure 4.1.

The main features include:

- The decentralized character of the overall MPC controller is such that each vehicle runs its own MPC scheme using the models and communicating only with its formation neighbors;
- Computational efficiency is achieved by replacing the (OCP) by a linear quadratic optimization problem (for which an efficient MATLAB solver is used) and, for this, we considered (i) quadratic cost functionals, (ii) approximation of each AUV dynamics by a linear model, and (iii) state constraints and control constraints (saturations) given by inequalities. The choice of this solver was motivated by the practicality of the future real control implementation onboard of the vehicle's computational system;
- Communication delays and packet dropouts can easily be incorporated; and
- Noise and disturbances can be easily incorporated in the simulated motion of the vehicles.

Incremental development

The complexity of the overall problem and the wealth of issues to be addressed recommend a step by step strategy to both research and development whereby issues

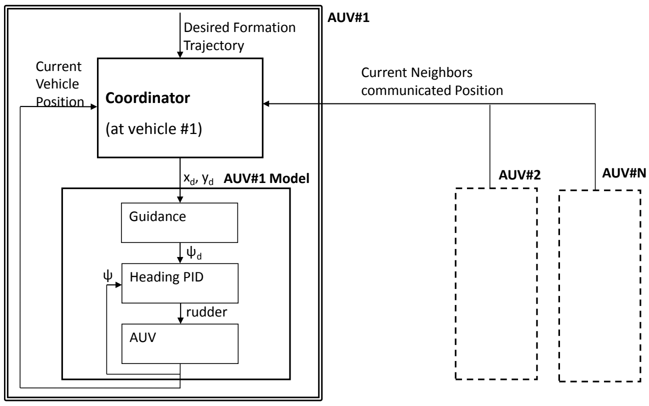


Figure 4.1: The overall simulation environment for AUV formation control

are considered, results are obtained, and solutions are implemented and tested incrementally. This allows a sustained development and a solid comparison with competing results.

In what concerns the development of a simulation framework, a first step is to provide a basis to implement and test the developments as well as to compare developments with what has been achieved in the current state-of-the-art. Although the general mathematical characterization of the decentralized formation control problem discussed in Section 4.2 is rigorous, its implementation based on the available control approaches entails a degree of computational complexity which is unrealistic in the light of the current AUV onboard computational and power capabilities. This conclusion is further reinforced from the typical large model uncertainty and relatively large magnitude of perturbations in the underwater milieu revealed by the large experience built on intensive field testing of AUV systems. Thus, it is not surprising that the following unicycle kinematic model

$$\dot{x} = v\cos(\psi), \quad \dot{y} = v\sin(\psi), \quad \dot{\psi} = u$$

has been widely considered in the AUV control literature. However, given the specific types of motion that will be considered in this first phase of the developments in which the navigation system is such that, for control purposes, the AUV can be regarded as a linear system on the plane, i.e., with dynamics given by $\dot{\xi} = A\xi + Bu$, where

$\xi = \begin{bmatrix} x & y & v_x & v_y \end{bmatrix}^T$, is the state (and output) variable, $u = \begin{bmatrix} u_x & u_y \end{bmatrix}^T$ and

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{\tau} & 0 \\ 0 & \frac{1}{\tau} \end{bmatrix},$$

where τ is some time constant and $u(t) \in U \subset \mathbf{R}^2$, being U a closed bounded set.

Moreover, by considering a discrete time version, we will be able to use very efficient off-the-shelf linear quadratic optimization solvers and compare the performance achieved in our developments with those in pertinent literature, as for example in (81, 87).

This simpler setup facilitates the analysis of the effects due to (i) perturbations entering additively in the vehicle's velocity, (ii) delays and loss of information due to acoustic communications, and (iii) control saturation in AUV models. These issues are being addressed in this order.

Certainly, this simpler setup also facilitates the addressing of key challenges inherent to the decentralized nature of the formation control problem. The crux of this matter lies in, under the tight communication, computation, and power constraints, finding a mechanism to enable the synthesis of a control "consensus" among AUVs (or groups of AUVs) on the basis of the overall connectivity while data exchanging of each vehicle is restricted to its neighbors. This constitutes a very broad research issue that will be listed in the next subsection.

We addressed the following set of formation control problems of an increasing order of complexity.

Firstly, we aconsider the case of several instances of a simple formation of two vehicles – one AUV and one simulated AUV, one ASV and one AUV, and two AUVs –, and investigate all associated control and technological issues as described above.

Then, more complex formations involving more than two vehicles will be considered. In what concerns implementation, the scenario of three vehicles will be addressed, being one of them an ASV in order to accommodate technological constraints, particularly those pertaining to the LBL positioning system. The cascaded leader-follower formation control problem is another scenario whose distribution of the burden of coordination among the AUVs for formation control requires a data flow that appears to be feasible.

However, the availability of an infra-structure for the localization of multiple AUVs is needed.

At each stage of complexity, this research encompassed the assessment of how the types of motion and communication perturbations impacts on stability, robustness, and sub-optimality.

In a later stage, we will considered more accurate AUV models and this entailed a MPC reformulation to overcome the computational limitations of the "on-the-fly" optimization.

The simulation developments were followed by the testing, fine-tuning, and validation of the control framework in the simulation environment but with real data from AUVs which was fed to the designed controllers, and finally, migrated to the vehicles' systems.

There are a number of remarks on research challenges that can be addressed in this framework:

- One issue concerns the computational complexity of "on-the-fly" optimization procedures with more realistic models of the vehicles. To address it, we considered a formulation of the MPC control scheme involving constraints specified by estimates of Attainable Sets. Besides the computational advantages, the new framework proved to be more versatile in what concerns the analysis of the effects of the various types of perturbations and uncertainties discussed above. Moreover, the additional insight proved to be useful for the analysis of the effects of control saturation.

- As pointed out above, this is a critical issue to extend the degree of decentralization of the formation control problem with the MPC control framework. The issue is to enable the efficient generation of a control consensus among vehicles on the basis of the overall connectivity while exchanging data only among the neighbors of each vehicle, in spite of the very tight constraints on communication, computation and power. Two avenues of research to address the issue of finding a way of enabling a feasible sharing of minimal information among the AUVs that suffices to fulfill the formation motion objectives and requirements and, thus, solving the formation control problem were considered. This constituted the basis for research that led for the following:

i) To cast the MPC scheme in a more efficient formulation involving adequate approximations to Attainable Sets and Value Functions.

ii) To consider specific topologies for the distribution of the coordination burden, like, for example, leader-follower, leaderless or a combination of both.

- In the context of *i*), we investigated conditions for the stability by seeking the assumptions on data of the problem under which the Value Function associated with the (OCP) satisfies the property of a Lyapunov function. First, the simple context of two vehicles will be considered and, then, generalized to more general and complex formations in which issues of string stability issues may be raised. In the later, we will seek to extend for our problem the approach for linear systems developed in (88) in which stability conditions are given in terms of the eigenvalues of the Laplacian matrix associated with the formation graph. Control-game theory and Attainable Set analysis concepts and results will be considered in order to investigate robustness and sub-optimality of the developed control structure.

- Another line of research addresses the challenges inherent to the control of a system over acoustic communications channels. Mechanisms to counter the negative impact of communications delays and loss of information in the stability and performance of the overall controlled system will be investigated. Replacement of the communicated data by simulated data, and adaptation of the MPC scheme parameters will be some of the ideas to consider. We will seek stability conditions for the controlled system with this modified MPC scheme. We will also examine the possibility of using the obtained conditions in order to define a feedback control system that ensure stability under almost minimal communications.

Now, we described the optimization based control synthesis that will be performed in each AUV as part of the overall decentralized MPC scheme implemented in the simulation environment.

Let N_p, n_v, and T be, respectively, the prediction horizon, the number of vehicles, and the sampling period (for now, assumed constant). Then, according to previous considerations, the discrete time linear model of vehicle $i = 1, \ldots, n_v$, is, for $k = 0, \ldots, N_p - 1$, given by:

$$x_{k+1}^i = \Phi^i(T)x_k^i + \Psi^i(T)u_k^i, \qquad y_k^i = C^i x_k^i \tag{4.4}$$

where $\Phi^i(T) = e^{A^iT}$, $\Psi^i(T) = \displaystyle\int_0^T e^{A^i(T-s)}dsB^i$, and $x_k^i \in \mathbb{R}^{n_s}$, $u_k \in \mathbb{R}^{n_c}$, and $y_k \in \mathbb{R}^{n_o}$ are respectively the system state, input and output variables, and n_s, n_c and n_o are the associated space dimensions. (Of course, it follows that $A \in \mathbb{R}^{n_s \times n_s}$, $B \in \mathbb{R}^{n_s \times n_c}$ and $C \in \mathbb{R}^{n_o \times n_s}$.)

From the considerations of the formation control problem formulation and assumed simplifications, it follows that the underlying (OCP) for AUV_i, starting at time t, involves data from all its neighboring vehicles as specified by the formation graph, and may be stated as follows:

$$
\begin{aligned}
(OCP_t^i)\ \text{Minimize}\quad & \sum_{k=1}^{N_p}(y_{t+k}^{ref,i} - y_{t+k}^i)^T Q^i (y_{t+k}^{ref,i} - y_{t+k}^i) + \sum_{k=0}^{N_p-1}(u_{t+k}^i)^T R^i u_{t+k}^i \\
& + \sum_{k=1}^{N_p}\sum_{j\in\mathcal{G}(i)}\Big(D^{ij}(y_{t+k}^i - y_{t+k}^j) - d^{ij}\Big)^T L^{ij}\Big(D^{ij}(y_{t+k}^i - y_{t+k}^j) - d^{ij}\Big) && (4.5)\\
\text{subject to}\quad & x_{t+k+1}^j = \Phi^j(T)x_{t+k}^j + \Psi^j(T)u_{t+k}^j, && (4.6)\\
& y_{t+k}^j = C^j x_{t+k}^j && (4.7)\\
& x_{t+k}^j \in [x_{LB,t}^j, x_{UB,t}^j] && (4.8)\\
& u_{t+k}^j \in [u_{LB}^j, u_{UB}^j] && (4.9)\\
& x_t^j = x_0^j, && (4.10)
\end{aligned}
$$

where the constraints have to hold for $j \in \{i\} \cup \mathcal{G}(i)$, with $\mathcal{G}(i)$ being the set of nodes of the graph specifying the formation that are connected with AUV_i, and for $k = 0, \ldots, N_p$.

Here, $\bar{y}_{t+k}^i = col(y_{t+k}^i, y_{t+k}^j; j \in \mathcal{G}(i))$ is the vector of all outputs of the pertinent vehicles, x_0^j is the initial state of vehicle j at the initial time t, D^{ij} is a matrix reflecting the formation relation between vehicles i and j, and d^{ij} is a parameter vector specifying the relation between vehicles i and j.

A compact representation of the constraints in this problem is obtained by considering: (i) $\bar{d}^i$, $\bar{x}_{t+k}^i$, $\bar{u}_{t+k}^i$, and $\bar{y}_{t+k}^{ref,i}$ defined as $\bar{y}_{t+k}^i$ above, and analogously, for the associated upper and lower bounds; and (ii) $\bar{Q}^i$, $\bar{R}^i$, $\bar{L}^i$, $\bar{\Phi}^i(T)$, $\bar{\Psi}^i(T)$, $\bar{C}^i$, and $\bar{D}^i$, respectively, the block diagonal matrices, formed with the Q^j, R^j, L^{ij}, $\Phi^j(T)$, $\Psi^j(T)$, C^j, and D^j, for $j \in \{i\} \cup \mathcal{G}(i)$ in the same order for all of them. Thus, the overall system equations can be written down exactly as in the statement of the above (OCP) but with the “bars” added and index i replacing j whenever it is the case.

We still need additional change of variables in order to formulate the linear quadratic optimization problem equivalent to (OCP_t^i).

Lets define the state and control vectors as follows

$$X^i = col(\bar{x}^i_{t+1}, ..., \bar{x}^i_{t+N_p}) \in \mathbb{R}^{n^i N_p} \text{ and } U^i = col(u_t, ..., u_{t+N_p-1}) \in \mathbb{R}^{m^i N_p}. \tag{4.11}$$

where the dimensions n^i and m^i follow from the previous constructions.

Then, the prediction model can be written as:

$$X^i = \mathcal{A}^i \bar{x}^i_t + \mathcal{B}^i U^i \tag{4.12}$$

where

$$\mathcal{A}^i = \begin{bmatrix} \bar{\Phi}^i(T) \\ \bar{\Phi}^i(T)^2 \\ \vdots \\ \bar{\Phi}^i(T)^{N_p} \end{bmatrix} \text{ and } \mathcal{B}^i = \begin{bmatrix} \bar{\Psi}^i(T) & 0 & \dots & 0 \\ \bar{\Phi}^i(T)\bar{\Psi}^i(T) & \bar{\Psi}^i(T) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\Phi}^i(T)^{N_p-1}\bar{\Psi}^i(T) & \bar{\Phi}^i(T)^{N_p-2}\bar{\Psi}^i(T) & \dots & \bar{\Phi}^i(T) \end{bmatrix}.$$

This prediction model computes the state trajectory X^i from a given initial condition $\bar{x}^i_t$ and a given control sequence U^i.

It is straightforward to conclude that state and control constraints are given by the inequalities

$$\mathcal{E}^i_x X^i \leq \mathcal{F}^i_x \quad \text{and} \quad \mathcal{E}^i_u U^i \leq \mathcal{F}^i_u$$

where $\mathcal{E}^i_x$, $\mathcal{E}^i_u$ are matrices, and $\mathcal{F}^i_x$, and $\mathcal{F}^i_u$ vectors of appropriate dimensions which can be defined as follows:

$$\mathcal{E}^i_x = \begin{bmatrix} I & 0 & 0 & \dots \\ -I & 0 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & -I \end{bmatrix} \quad \mathcal{F}^i_x = \begin{bmatrix} \bar{x}^i_{UB} \\ \bar{x}^i_{LB} \\ \vdots \\ \bar{x}^i_{UB} \\ \bar{x}^i_{LB} \end{bmatrix} \quad \mathcal{F}^i_u = \begin{bmatrix} \bar{u}^i_{UB} \\ \bar{u}^i_{LB} \\ \vdots \\ \bar{u}^i_{UB} \\ \bar{u}^i_{LB} \end{bmatrix}$$

and $\mathcal{E}^i_u$ as the same structure as $\mathcal{E}^i_x$, differing only in the dimension.

Now and just like in (4.11), we define Y^i, $Y^{ref,i}$ and $\mathbf{d}^i$ from, respectively, $\bar{y}^i_{t+k}$, $\bar{y}^{ref,i}_{t+k}$ and $\bar{d}^i$, as well as the block diagonal matrices $\mathcal{Q}^i$, $\mathcal{R}^i$, $\mathcal{L}^i$, $\mathcal{D}^i$, and $\mathcal{C}^i$, respectively, formed with $\bar{Q}^i$, $\bar{R}^i$, $\bar{L}^i$, $\bar{D}^i$ and $\bar{C}^i$. We are now ready to formulate a linear quadratic programming problem equivalent to (OCP_t^i)

$$\begin{aligned} (LQOCP_t^i) \text{ Minimize} \quad & U^T H^i U + 2 f^i U \\ \text{subject to} \quad & A^i_c U \leq b^i_c \end{aligned}$$

$$\begin{aligned}
\text{where } H^i &= \mathcal{B}^{i^T}\mathcal{C}^{i^T}\mathcal{Q}^i\mathcal{C}^i\mathcal{B}^i + \mathcal{B}^{i^T}\mathcal{C}^{i^T}\mathcal{L}^{i^T}\mathcal{Q}^i\mathcal{L}^i\mathcal{C}^i\mathcal{B}^i + \mathcal{R}^i, \\
f^i &= \mathcal{B}^{i^T}\mathcal{C}^{i^T}\left[\mathcal{Q}^i(\mathcal{C}^i\mathcal{A}^i\bar{x}_t^i - Y^{ref,i}) + \mathcal{D}^{i^T}\mathcal{L}^i(\mathcal{C}^i\mathcal{A}^i\bar{x}_t - \mathbf{d}^i)\right] \\
\text{with} \qquad & A_c^i = \left[\begin{array}{ccc} \mathcal{E}_u^i & \mathcal{E}_x^i & \mathcal{B}^i \end{array}\right] \quad \text{and} \quad b_c^i = \left[\begin{array}{c} \mathcal{F}_u^i \\ \mathcal{F}_x^i - \mathcal{E}_x^i\mathcal{A}^i\bar{x}_t \end{array}\right].
\end{aligned}$$

This optimization problem can be solved using efficient quadratic programming solvers. For instance, *quadprog* function in available in the Matlab Optimization Toolbox.

One can immediately conclude that any reasonably small original (OCP) can lead to an optimization problem of large dimensions. It is well known that the computational complexity is proportional to $(m_i + N_p)^3$.

4.5 MPC scheme

In this section, we describe the currently implemented version of the MPC scheme for the control of a formation of AUVs. This scheme runs in each vehicle and, in this first implementation, will be the same for all the vehicles. Thus, if there is no loss of information in the communication, then, all the vehicles share the same data and the control strategy generated for each vehicle is known by all of them. In the event of packet dropouts or communication delays, the missing sampled data is replaced by simulated data, and there will be some differences between the control strategies computed by the various vehicles for a given vehicle. As it will be seen in the simulation results, these differences are relatively small but with a noticeable effect in the loss of performance.

The implemented MPC scheme in AUV_i is as follows:

1. Initialization: Setting of prediction and control horizons, and of other (OCP) parameters that depend on the specific mission requirements, such as, level of perturbations, existence of obstacles, relative importance of trajectory tracking and formation pattern errors.

2. Sample the state variable, compute its estimate by applying a Kalman filter, and communicate this estimate to its neighbors via acoustic modem.

3. Obtain the state variable of its neighbors via acoustic modem.

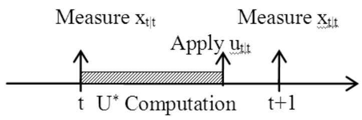


Figure 4.2: The MPC scheme time line

   (a) If data is available, goto step 4.

   (b) Otherwise, generate the neighbors' state variables obtained by running their models.

4. Solve the linear quadratic optimization problem ($LQOCP_t^i$) at the current time t for the current prediction horizon (of length N_p) and the given reference output trajectory, $[\bar{y}_{t+1}^{ref,i}, ..., \bar{y}_{t+N_p}^{ref,i}]$.

   This yields the optimal control sequence $[u_t^{i*}, ..., u_{t+N_p-1}^{i*}]$ (and, of course, the corresponding output trajectory, $[y_{t+1}^{i*}, ..., y_{t+N_p}^{i*}]$) for vehicle i.

5. Apply the control u^{i*} for the current control horizon.

6. Slide time for the optimization problem and adjust parameters if needed

7. Let time elapse until the end of the current control horizon, and goto step 2.

The relation between the computation and control application times can be examined in figure 4.2.

We ran this scheme in an environment context, for small formations of two and three AUVs. We believe that this MPC scheme can be applied for formations of the type leader-follower, or even cascaded leader-follower, which exhibit a pattern of distribution of the coordination burden compatible with the onboard resources constraints.

However, given the tight constraints of our application scenario (discussed in the previous section), this scheme as described above can not be generalized for more complex formations with a large number of AUVs in which each one communicates acoustically only with its neighbors. Thus, since each vehicle uses only partial information that might be different from the data of any other vehicle (in particular, of any of its neighbors), conflicting control strategies may arise and mechanisms to generate consensus are required. Thus, it is extremely difficult, if not impossible, to implement a realistically valid operational system due to the incompatibility of the very low resources budget available onboard each vehicle, and the communication constraints with the required number of data exchange iterations.

4.6 Simulation results

Communication model

Communication is an important part of a networked system, and, the more so in the underwater milieu since, generally, the communication channel is acoustic, and, therefore, exhibits very low data rate transmission, and an unreliability level which are extremely challenging from the control point of view.

The communication model reflects its impact in the information flow among the AUVs. These are essentially of two types:

- Transmission delay in the communication channel due to the fact that sound propagates in the water at a speed of approximately 1500 m/s as well as to a certain latency time that depends on the specific acoustic modem. This delay is easily estimated from the estimate of the distance between the two vehicles. The sound speed varies slightly with the water salinity and temperature but, at this time, we will not consider such effects.

  Further delays may also occur at the receiver if the acoustic wave propagation takes a longer path which may due to, for example, multiple-path reflections. However, we will not consider this possibility. Data is time stamped and if it does not arrive approximately within a certain time interval centered on the estimated delay, then it will be discarded and the corresponding information packet is considered lost.

- The loss of data packets (packet dropouts) is an important feature to be considered as the perturbations of the underwater environment are quite significant.

Both these features have a very important impact in control: Delays may lead to instabilities and the loss of data in the communication channel implies that the system will be simply in open loop. Thus the control system has to be prepared to take into account these issues. This is a point for which the redundancy of the controls computed in the MPC scheme can be exploited. This redundancy is due to the fact that the prediction horizon is, usually much larger than the control horizon. So, if fresh data fails to be received, than previously computed controls can be used for the elapsing time slots until the next successful communication event happens. In the meantime, models of the other vehicles can be used in order to compute estimates of their state variable evolution during the time period of interest.

In what concerns the structure of the information packet to be sent to the neighboring vehicles required by the basic MPC scheme – i.e., the one in which each vehicle solves identical optimization problems –, this includes only the state variable estimate obtained after filtering (with appropriate Kalman filter) the sampled state variable and its time stamp.

However, for more general schemes, in which the set of neighbors of each one of the communicating vehicles differ, it is of interest to send either control or state variable values at each one of the time instants of one or more of the control horizons. This will be useful to ensure robustness with respect to future data packet dropouts.

If several samples of a given variable at different times are to be transmitted, then the implementation of the communication model might encompass a linear data buffer at the receiver end. These samples are ordered by the time they refer to. Every time a data set is removed from the buffer, the remaining data sets are shifted one position. On the other hand, every time a data set is transmitted by a vehicle, the entry position reflects the time at which the variable refers to in order to ensure that will be removed from the buffer at the right time instant.

Noise and disturbance analysis

In the current stage of development of the simulation environment, noise is incorporated in two ways: vehicles' models and communications.

To simulate the vehicle's sampled data more realistically and thus, enable to test the robustness of the designed controllers, Gaussian noise, v_k and s_k, with mean and variance as choice parameters, are added to the vehicle's dynamics as an additional input and to the output sensor readings, respectively, that is,

$$x_{k+1} = Ax_k + Bu_k + v_k, \text{ and } y_k = Cx_k + s_k. \tag{4.13}$$

Then, of course this data is used to compute the best estimate of the state variable values, by using the usual Kalman filter as follows:

$$\begin{aligned}
\text{Prediction step:} \quad & \hat{x}(k|k-1) = A\hat{x}(k-1|k-1) + Bu(k-1) \\
& P(k|k-1) = AP(k-1|k-1)A^T + Q \\
\text{Update step:} \quad & \hat{x}(k|k) = \hat{x}(k|k-1) + K(k)e(k) \\
& P(k|k) = P(k|k-1) - K(k)SK(k)^T \\
\text{being,} \quad & e(k) = y(k) - C\hat{x}(k|k-1) \\
& K(k) = P(k|k-1)C^T[CP(k|k-1)C^T + R]^{-1}
\end{aligned}$$

where $\hat{x}(k|k)$ and $\hat{x}(k|k-1)$ are the state estimate at time k given, respectively, all available measurements, and the first $k-1$ measurements (the later is also called the state prediction), similarly, for $P(k|k)$ and $P(k|k-1)$ for the error covariance matrix, $e(k)$ the innovation, $K(k)$ the Kalman gain, Q and R are, respectively, the process and sensor noise covariance matrices, and, finally, $y(k)$ and $u(k)$ are, respectively, the output and the control input variables.

4.7 Hardware-in-the-loop simulation results

General considerations

In this section, we present results obtained with the developed simulation environment in which the MATLAB linear quadratic programming solver is used. This framework exhibits the following features:

- Quadratic cost functions weighting the reference trajectory tracking error, control effort, and the formation pattern error.

- Control systems with linear dynamics and subject to noise of the Gaussian type with “adjustable” mean and variance, added as an additional input in the vehicle dynamics. Once sampled the state variable, a Kalman filter is used to obtain a state estimate to be used by the optimization solver and communicated to neighboring vehicles.

- Control constraints enabling the consideration of saturations.

- State/output inequality constraints (obstacle avoidance). These enable the incorporation of obstacles and the assessment of the performance of the proposed MPC scheme with obstacle avoidance.

- Communication model. Communicated data is time stamped and may exhibit a time delay proportional to the distance between the vehicles exchanging data or subject to packet dropouts. If a given packet of information is not received within a time window centered around its expected delay, then a dropout is assumed. Thus, the performance sensitivity of the MPC controller can be assessed with respect to either or both time delays and packet dropouts. Each vehicle has a linear buffer enabling it to receive multiple data samples from other vehicles and whose implementation is described in the previous section. In the current simulation experiments, packet dropouts have been considered in the stochastic context with gaussian model.

- The following performance metrics - exemplified for the case of two vehicles - are being considered:

  - TM - Tracking Metric - The Euclidean norm of the reference trajectory tracking error (corresponding to L_2 norm in continuum time) - it measures how far the AUVs are from the trajectory to be tracked and is given by

$$TM = \frac{TM_1 + TM_2}{2}, \text{ where}$$

$$TM_i = \sqrt{T \sum_{k=1}^{N_p} \left[(x_{1,k}^{ref,i} - x_{1,k}^i)^2 + (x_{2,k}^{ref,i} - x_{2,k}^i)^2 \right]}, \quad i = 1, 2.$$

  - FM - Formation Metric - The Euclidean norm of the formation pattern error (corresponding to L_2 norm in continuum time) - it measures how far the formation is from their formation pattern. Here, we consider a formation defined by (i) a constant lateral distance d, and (ii) the vehicles should travel side by side. . It is given by:

$$FM = \sqrt{T \sum_{k=1}^{N_p} \left[(x_{2,k}^1 - x_{1,k}^1 - d^1)^2 + (x_{2,k}^2 - x_{1,k}^2 - d^2)^2 \right]},$$

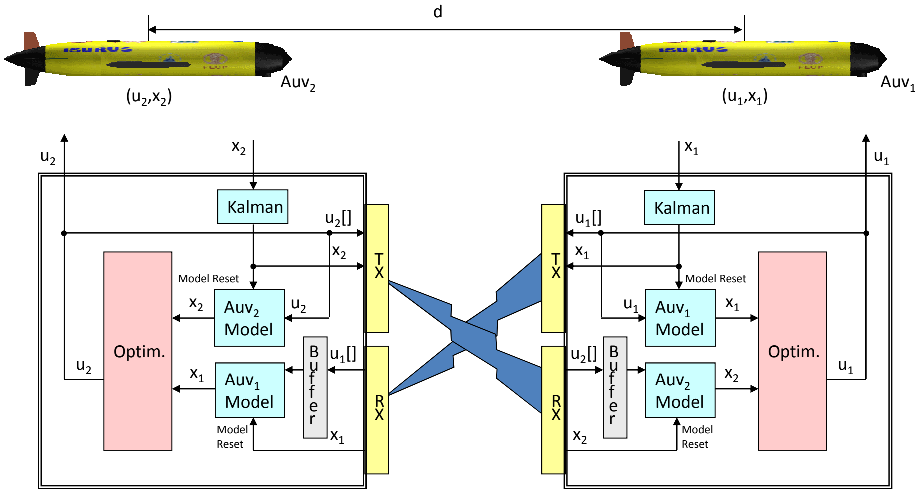


Figure 4.3: Simulation of MPC scheme for a two AUV formation

where $\bar{d} = col(d^1, d^2)$ with $|\bar{d}|_2 = d$ is a vector pointing from AUV_1 to AUV_2 orthogonal to the average of the vectors tangent to the short term time averaged paths being followed by both AUVs.

- CE - Control Effort - The control effort given by the Euclidean norm of the control function and is given by

$$CE = \sqrt{T \sum_{k=1}^{N_p} |u_k|^2}.$$

- The cost functional adopted in the MPC synthesis evaluated along the whole simulation time horizon.

These measures provide a complete assessment of the controller's performance.

Results

At this stage of research, we consider only very simple formations that served to assess the simulation framework as well as to provide some initial insight into the challenges that we are addressing later.

The first batch of data concerns a simple formation of two AUVs that have to track their trajectories, and, at the same time, travel side by side while maintaining a constant distance between them. Figure 4.3 shows the setup for the control of a formation with two vehicles.

A prediction horizon of 5 time steps, a total time horizon of 9 seconds, and a sampling time of 0.1 seconds were defined, being the cost function weights given by:

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \qquad L = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \qquad R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

These weights were tuned in order to obtain a good compromise between trajectory tracking and formation keeping.

While, the first one penalizes the tracking error to a sinusoidal reference trajectory, the second penalizes the formation error and the last one the control effort. The lateral distance d between the vehicles characterizing the formation pattern is measured along the direction given by the average of the tangents to both vehicles' trajectories and it can be varied in order to assess the impact of the communication channel delay in the performance of the MPC controller. The nominal velocity is of 1 m/s and the control input is allowed to take values between -10 and $+10$

We also considered a square shaped obstacle, $\mathcal{O}$, intersecting the reference trajectories that the vehicles are supposed to track so that they are forced to circumvent it in order to avoid a collision. The following situations were considered.

In the simulation experiments, we plot the motion of a simulated AUV without perturbations and subject to the inputs generated by the MPC controller by taking into account the effect of perturbations, as depicted in figure 4.4. This conveys a very good idea of the effectiveness of the MPC controller in coping with the effects of perturbations.

Gaussian noise with variable mean and variance was added as an input to each of the vehicles, and, for a more realistic simulation, a Gaussian noise of zero mean and variance of 0.16 was considered for the output sensor for all the situations except the "deterministic" scenario. An average of ten sample runs with independent noise was obtained for each situation in order to assess the MPC controller performance.

The performance of the controller was assessed through the above mentioned four indicators: Euclidean norm of the trajectory tracking error (TM), Euclidean norm of the formation error (FM), "L_2" norm of the control function (CM), and the value of the cost functional evaluated along the control process (the pair of control and trajectory) evaluated during the whole time horizon (C).

Trajectory, metrics and the optimization cost are available to evaluate, in a simulation context, the system performance in the following situations:

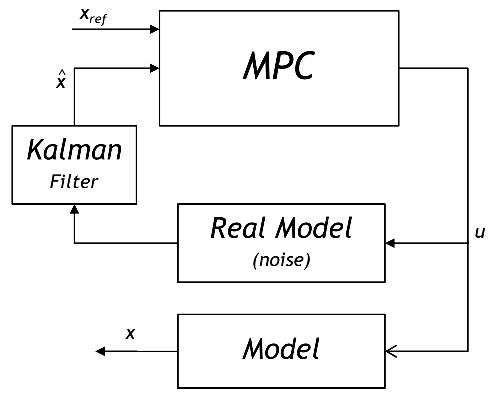


Figure 4.4: Scheme for a qualitative assessment of the MPC effectiveness

- Comms Off. In this scenario of no communications between the two AUVs, each vehicle runs its own optimization problem without data being communicated by the other vehicle. However, an estimate of the initial state of the other vehicle is known and simulation data of the state evolution of the other vehicle is used.
- Comms On and no delay. Each vehicle runs its own optimization problem with data being communicated by the other vehicle without any delay.
- Comms On and delay of 0.1 seconds. Each vehicle runs its own optimization problem with data being communicated by the other vehicle with a delay of 0.1 seconds. The prediction model was used to estimate the other vehicle's position in order to compensate for the delay.

In these three situations, the input noise was always Gaussian with the following seven levels, being the mean and variance considered componentwise:

Mean	0	0	0	0	0	0.1	0.2
Var.	0	0.02	0.05	0.1	0.25	0.02	0.05

After running the simulation in each one of the above conditions, the obtained results are presented in Table 4.2.

Associated with some of entries of Table 4.2, we include a number of graphs of the trajectories of specific runs to illustrate the discussion. In these:

Table 4.2: MPC controller performance table

Noise Level Situation	Mean Var.	0 0	0 0.02	0 0.05	0 0.1	0 0.25	0.1 0.02	0.2 0.05
Comms = Off	TM =	0.75	1.24	3.17	11.78	33.51	211.72	207.64
	FM =	0.19	0.62	1.42	2.87	4.85	39.60	77.07
	CM=	8.21	15.51	27.59	40.63	48.28	57.79	58.81
	C=	34.39	87.86	206.60	524.90	1158.00	8197.00	11862.00
Comms=On	TM=	0.75	0.76	0.79	0.83	1.02	1.11	2.29
Delay=0	FM=	0.19	0.21	0.25	0.31	0.49	0.47	0.84
	CM=	8.21	8.85	10.60	14.70	25.90	17.65	29.53
	C=	34.39	36.40	41.59	48.45	70.30	81.34	157.61
Comms = On	TM=	0.75	0.77	0.80	0.88	1.27	1.67	3.59
Delay = 0 .1	FM=	0.19	0.22	0.27	0.34	0.48	0.82	1.61
	CM=	8.21	10.92	16.05	24.51	34.71	18.36	33.13
	C=	34.39	37.54	44.19	52.55	74.95	105.54	208.25

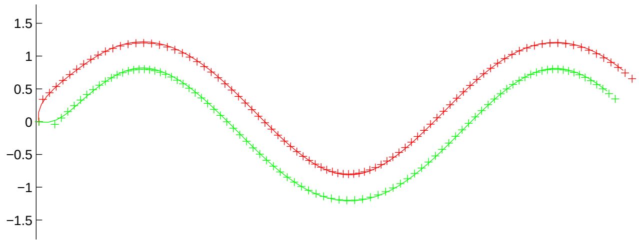
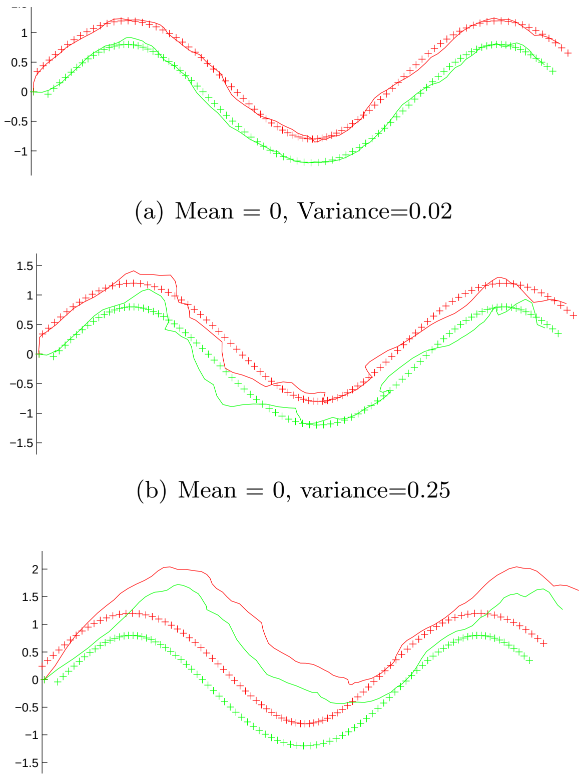


Figure 4.5: Formation trajectories without AUV communications, noise or delay

- red refers to AUV_1, and green refers to AUV_2;
- "+" represents the given trajectory reference to be tracked;
- solid line represents the actual trajectory of the AUV (model affected by disturbances) feedback with control generated by the MPC system; and
- whenever present, "o" represents the real position when the control of the previous system is applied to the model without perturbations. This graph acts as a reference to show how good is the controller countering the effect of disturbances and delays.

A close inspection of the table reveals the following remarks:

- It is not surprising that, in the deterministic case (no input noise and no output sensor noise), the considered three different situations yield the same performance. See figure 4.5.
- In all the three situations, all the four performance criteria worsened with the increase of the noise level. This can easily be concluded by inspection of the trajectory graphs in figure 4.6 However, one has to acknowledge that:
  - The impact of the mean increase is much greater than that of the variance.
  - A comparison between the situations Communications Off and On, reveals that the MPC controller with communications between both vehicles is extremely effective in softening the effect of noise in all the criteria. Moreover, as it can be seen in figure 4.7, the lack of communication between vehicles implies a very poor performance not only in formation keeping, but also in trajectory tracking.



(a) Mean = 0, Variance=0.02

(b) Mean = 0, variance=0.25

(c) Mean = 0.2, Variance=0.1

Figure 4.6: Formation trajectories with AUV communications, and increasing noise levels

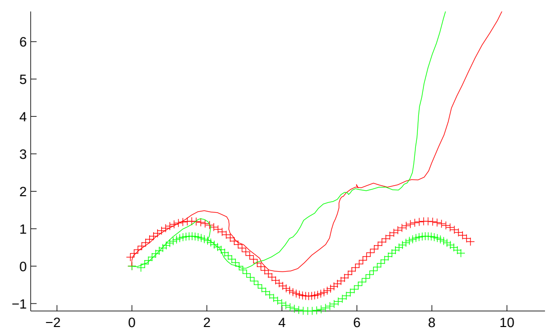
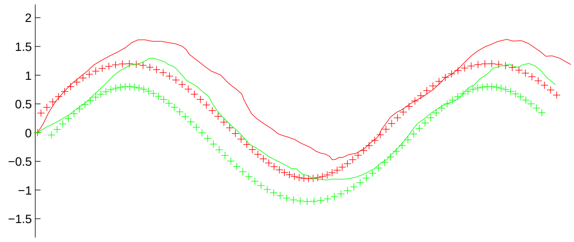
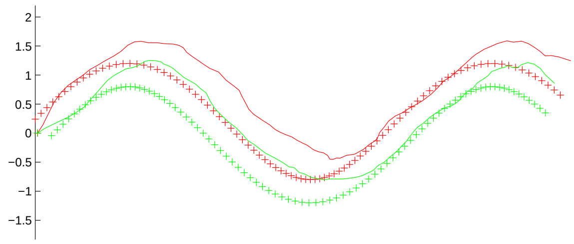


Figure 4.7: Formation trajectories without AUV communications, and Gaussian noise with mean and variance equal to $(0, 0.1)$



(a) No delay



(b) delay of 0.1 sec compensated by predictor

Figure 4.8: Formation trajectories with AUV communications, Gaussian noise with mean and variance equal to $(0.1, 0.05)$

- There is not much difference in what concerns the noise sensitivity of the situations without and with delay compensated by a predictor. This can be seen both in the above table and graphically by comparing both trajectory graphs in figure 4.8. This is not surprising since the predictor is used to counter the delay, leaving only the effect of noise in the last sampling period "uncompensated". However, this is not the case, when the mean of the noise becomes nonzero

- To appreciate the impact of the MPC controller, see figure 4.9. The trajectory marked by "o" is that an AUV not subject to noise when moving with a control generated by the MPC for a vehicle subject to Gaussian noise with zero mean and variance 0.1 (whose trajectory is depicted with a solid line).

The significance of the effect of the predictor in canceling the delay can be clearly seen in Table 4.3 below (generated in the same way as above). There is a reasonable performance improvement due to the inclusion of a predictor that partially cancels the effect of the delay. This improvement is not easily detectable in the realization of the two formation trajectories with identical noise depicted in figure 4.10.

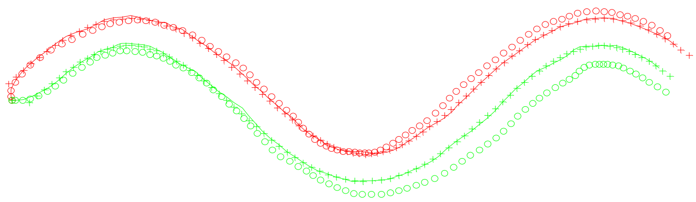
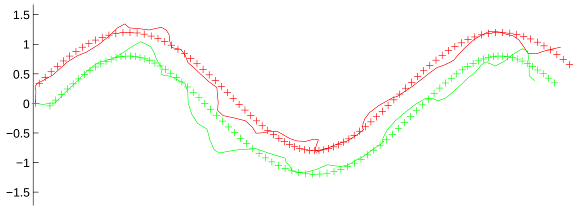


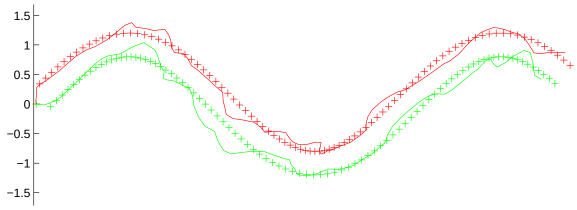
Figure 4.9: The effectiveness of the MPC based controller

Table 4.3: Effect of the predictor in the MPC controller performance

Noise Level Situation	Mean Var	0 0	0 0.02	0 0.05	0 0.1	0 0.25	0.1 0.02	0.2 0.05
Comms On Delay=0 .1 Estimator On	TM= FM= CM= C=	0.75 0.19 8.21 34.39	0.77 0.22 10.92 37.54	0.80 0.27 16.05 44.19	0.88 0.34 24.51 52.55	1.27 0.48 34.71 74.95	1.67 0.82 18.36 105.54	3.59 1.61 33.13 208.25
Comms On Delay=0 .1 Estimator Off	TM= FM= CM= C=	1.32 0.37 13.87 63.77	1.32 0.39 14.93 66.43	1.34 0.43 17.56 71.17	1.38 0.49 22.53 77.58	1.56 0.65 31.55 97.31	1.61 0.71 19.22 104.08	3.33 1.09 27.95 177.50



(a) With predictor to compensate delay



(b) Without predictor to compensate delay

Figure 4.10: Formation trajectories with 0.1 sec communications delay, and Gaussian noise with mean and variance equal to $(0, 0.25)$

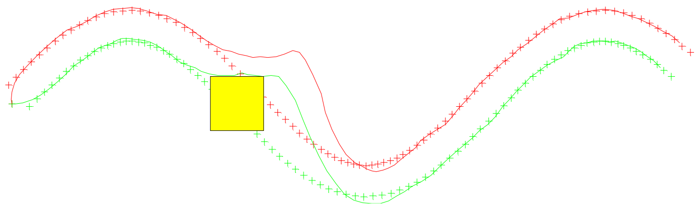


Figure 4.11: Obstacle avoidance with an MPC based controller

Figure 4.11 illustrates the versatility of the MPC based controller by enabling the incorporation of obstacle avoidance with minimal formation degradation. As explained earlier, this situation was achieved by including additional state constraints, not initially considered in the optimization problem, at some point in time at which an obstacle was detected. The MPC optimization problem produces a trajectory for both vehicles which avoids collision with the obstacle, and at the same time, keeps on minimizing the original cost functional.

The MPC scheme was also implemented for formations of three vehicles, and the results observed with a somewhat limited simulation experience, corroborate the ones observed for formations with two vehicles for the case in which all the vehicles communicate with each other. The other two situations in which the vehicles communicate pairwise or one of the them communicates with the other two but these do not communicate between them is still under research. This is a scenario in which decentralization issues arise and, as pointed out earlier in this work, important challenges have to be addressed in order to cope with the strict constraints of the underwater environment.

The simulation results of an implementation of a MPC based controller simple formations of AUVs presented and discussed here reveal that, as expected, performance worsens with the level of noise and, more significantly, with the delay. One also concludes that the overall performance is extremely sensitive to the cost functional weights, particularly the one of the control. However, the MPC based controller exhibits a very good robustness to input noise and output sensor noise – in that the performance degrades very slowly with the increase of variance – specially for the case in which the mean of the noise is very small. Another important feature concerns the resilience to small delays. The simulation experience also reveals the large sensitivity of the overall performance with respect to the cost functional weights, particularly the one of the control. The exploration of this issue having in mind the definition of easy control design guidelines is the subject of near future research. The issues addressed here will be compounded in complexity if more complex formations are considered. Although, new issues like, for example, stability, extent of decentralization, and tractability will definitely require further research and novel developments, we believe that the simulation environment presented here constitutes an excellent tool to support the required research.

4.8 Conclusions

In this chapter, conventional MPC scheme was designed, implemented, and tested in two different instances: single AUV, and a decentralized triangle formation of AUVs (with one leader and two follower) whose mission consists in tracking a given path while avoiding the collision with unexpected obstacles. A linear quadratic (OCP) with both state and control constraints wa considered in the designed MPC scheme. The MPC scheme generated appropriate waypoints which then fed the low level controllers. The testing process involved two phases: simulation, and simulation with real AUV hardware-in-the-loop with real field data. A commercial powerful quadratic programming solver was used in the real-time implementation.

The assessment is an upside and a downside. The positive aspects is that, in both cases (single AUV and AUV triangle formation) and in both contexts - software and hardware-in-the-loop simulations, a good performance was achieved - both in tracking error, fuel consumption, and obstacle collision avoidance - in various reasonable standard real life operational situations in what concerns the level of motion disturbances and communications reliability. It should be noted that, in the experiments, the triangle formation and the role of the AUVs were preserved during the maneuver to avoid the collision with unexpected obstacles. The AUV hardware-in-the-loop experiments also revealed that the consumption of onboard power and computational resource were not difficult to accommodate for missions with an endurance considered typical for the class of AUVs used. The negative side is that, in spite of the proved viability of the motion requirement, it became also clear that the computation, communications, and power budget for the motion and navigation control was, with the current technological state-of-the-art hardware, quite significant, leaving relatively small and inflexible room for the payload activities which are, in fact "la raison d'tre" of the overall system. Moreover, if the motion flexibility – in terms of range of maneuvers, set of underwater milieux states – of the set of missions to be considered and the number of vehicles were to be increased, then the motion control system addressed in this chapter would be clearly unsatisfactory in what concerns fulfilling reasonable mission requirements.

It is a fact the control system implementation could still be optimized but the extra resources made available would not suffice to ensure the competitiveness hedge in the context of the challenges that lie ahead in the near future for these systems. This points out to the need of radically new control frameworks that allow the combination of feedback control while optimize the scarce on-board resources.

Chapter 5

The Attainable Set Model Predictive Control Scheme

5.1 Introduction

In this chapter, we introduce a new formulation of the Model Predictive Control (MPC) scheme having in mind to reduce as much as possible the on-line computational burden present in the conventional schemes. This feature is particularly important to increase the range of applications exhibiting severe real-time constraints. Moreover, since running complex optimization algorithms typically requires significant power consumption, the novel MPC scheme also mitigates the loss of endurance when power hungry optimization algorithms are required for the control synthesis.

The developments and results of Chapter 4 clearly shows how challenging is the problem of controlling the motion of formation of multiple AUVs in a coordinated fashion. Thus, it serves as a strong motivation to showcase the novel proposed scheme. It will be clear from the developments of this chapter, that the new MPC scheme deals well with:

(a) Modeling uncertainties, motion perturbations, environment variability, and emergence of obstacles; and

(b) Performance optimization requirements subject to a number of very diverse type of constraints, for which the versatility of the optimal control paradigm is particularly well suited.

These two general classes of issues justifies the enormous amount of research undertaken in the past, and that led to a wide variety of MPC schemes, (81, 85, 86, 96, 98, 107), among others. In these works, diverse variants of the following basic MPC scheme have been adopted:

i) Initialization;

ii) Sampling the state of the vehicles and computing an estimate of their state variables;

iii) Sharing the data obtained in ii), generally via acoustic communication;

iv) On-line solving in each AUV of an optimal control problem which encompasses data from neighboring vehicle;

v) Each vehicle applies the computed control strategy during the control horizon which, typically, is a short fraction of the prediction horizon; and

vi) once the control horizon interval elapses, the new prediction horizon is considered by appropriately sliding time, and the procedure re-initiated in step ii) for the new current time.

Obviously, the price to pay for the long term optimization quest is the computationally intensive character of the control synthesis: optimal control problems are notoriously known for being computationally difficult, (108, 109, 110, 111). In order to cope with this, the developments reported in chapter 4, which follow along the ones in (81), an MPC control scheme for the coordinated control of a formation of AUVs based on a linear quadratic optimal control problem was adopted. This formulation is particularly useful because it brought together the computational advantages of existing extremely efficient numerical solvers and the flexibility exhibited by the conventional optimal control problem which enables the incorporation of a wide range of control and state constraints arising in the control of AUV formations.

Unfortunately, the implementation of sophisticated requirements - which are one the most welcome features of the MPC control framework - comes with a very high computational complexity which is a prohibitively high price - even when piece-wise affine approximations to the ingredients (functions and sets) of the associated (OCP) are used -, particularly for applications involving a large number of vehicles subject to real-time requirements under very strict constraints. At the core of the conventional MPC scheme there is the need to solve the computationally expensive (OCPs) after each (usually short) control period has elapsed. At each step of this optimization process, the integration of a set of, often complex, differential equations over the optimization horizon, is required.

This translates in an awful waste of computational effort relatively to an approach that avoids repeating computations that involve time-invariant data in an essential way. Chiefly among these, are those representing the dynamics of the AUV(s). These equations can be integrated off-line, as a function of the value of the state variable at the initial time, over the control horizon for an appropriate set of control functions by taking into account all pertinent time-invariant data (a priori known obstacles, currents, etc.). Of course, along with this data, the current control horizon final time equivalent to the optimization horizon cost functional needs to be computed.

For the usual situation in which the long term optimization is of interest, an approximation to the Value Function is a convenient object. Its computational burden is very heavy but it can be computed off-line for the points in the state space of interest by considering a priori known time-invariant data. It is important to remark here that, in case of detection of an unexpected event - say, emergence of an unanticipated obstacle or underwater current - during the control horizon, the update of the Value Function is only required in a certain region of the state space in which a modified control action has to be exerted until the influence of the unexpected event in the optimal behavior of the vehicle becomes negligible or even null.

These ideas are an informal and general outline of the novel MPC approach presented in this chapter.

Here, we propose an MPC-like control that substantially reduces the computational burden associated with the conventional MPC scheme, even for highly nonlinear dynamics. Its main features consist in:

1. Replacing the optimization problem over the control space by another one over a local approximation to the Attainable Set, i.e., the subset of the state space that can be attained at the end of the control horizon by using all feasible control functions.

2. Propagating the long term cost functional to the final time of the current control horizon, via the associated Value Function.

3. Solving the optimization problem for the control synthesis over the control horizon which is much shorter than the control horizon considered in the usual MPC schemes.

A few observations are in order:

First, the Attainable Set and the Value Function can be computed or, at least, approximated, off-line by taking into account all the (state and control) constraints as a function of the initial state.

Second, the fact that we consider approximations to the Attainable Sets facilitates the incorporation and handling of geometric constraints when solving optimization problems. In particular, very fast optimization solvers based in very efficient search algorithms can be used.

Third, the difficulties inherent to the linearization of the control system dynamics, either by first order approximation with its associated control issues, or by lower level feedback control deeply compromising the overall optimization, are avoided.

Fourth, uncertainties and perturbations can be dealt with by considering either mini-max optimization schemes where the synthesized control optimizes the worst case due to their potential effect, or, by paying a small sub-optimality price, consider a number of control horizon intermediate steps just to correct the effect of small persistent disturbances.

This chapter is organized as follows. In Section 5.2, we start with the conventional MPC scheme presented in Section 3.4 and focus on previous research approaches to overcome the inherent computational complexity, particularly, in the presence of real-time constraints and limited computational capabilities.

Then, in Section 5.3, the basic Attainable Set MPC (AS-MPC) scheme is presented and its equivalence to the conventional scheme justified. Convergence properties are presented and proved for this general abstract framework. Relatively, to the conventional MPC, the new proposed scheme has the advantage of transferring the heavy on-line computational burden of solving the optimal control problem to an off-line stage by taking advantage of the time invariance of the dynamic system and of constraints. Also in Section 5.4 we will address optimality, asymptotic stability and robustness properties of the AS-MPC scheme.

However, the abstract scheme is still plagued by a very significant computational burden. Indeed, computing the Attainable Set of a dynamic control system is still an extremely demanding computational process. Thus in Section 5.5, we present and compare three approaches - polyhedral, ellipsoidal, and "cloud of points" - to approximate the Attainable Set and provide the justification to select a specific version of the last one. Some results providing estimates on the Hausdorff distance between the Attainable Set and its approximation are presented. This section is completed with optimality, and stability results, as well as, discussion on robustness, for the AS-MPC scheme for the case in which the Attainable Set is replaced by its approximation.

Finally, a robust version of the proposed AS-MPC (RAS-MPC) is presented and discussed. The envisaged scenario consists in eliminating small and persistent perturbations that might prevail within each control horizon. The idea consists in closing the loop within the control horizon in order to compensate for the perturbations effects. For this, the optimality at each step has to be sacrificed to small extent.

This chapter is closed with some conclusions and open issues.

5.2 From the conventional MPC to the Attainable Set MPC

Let us restate, for convenience but also with more detail, a common version of the conventional MPC scheme presented in section 3.2 which has been considered in a number of seminal publications addressing a wide spectrum of important issues such as stability, sub-optimality, robustness, decentralized schemes, etc., e.g., (72, 73, 78, 79, 81, 86, 95, 97) and references cited therein.

1. Initialization. Let t_0 be the current time, and set up the initial parameters or conditions specifying the initial state, prediction horizon and control horizon, respectively, x_0, T, and Δ, and, possibly other parameters.

2. Sample the state variable at time t_0.

3. Compute the optimal control strategy, u^*, in the prediction optimal, i.e., $[t_0, t_0 + T]$, by solving the optimal control problem:

$$
\begin{aligned}
(P_T) \text{ Minimize} \quad & g(x(t_0+T)) + \int_{t_0}^{t_0+T} f_0(t, x(t), u(t))dt \\
\text{subject to} \quad & \dot{x}(t) = f(t, x(t), u(t)), \ \mathcal{L}\text{-a.e.} \\
& u(t) \in \Omega \quad \mathcal{L} - a.e. \\
& h(t, x(t)) \leq 0, \ \forall t, \ x(t_0+T) \in C_f,
\end{aligned}
$$

   where g is the endpoint cost functional, f_0 is the running cost integrand, f, h, and g represent, respectively, the control system dynamics, the state constraints, and the mixed constraints, C_f is a target set which may also be specified in order to ensure stability.

4. Apply the obtained optimal control during the current control horizon, $[t_0, t_0+\Delta]$.

5. Slide time by Δ, i.e., $t_0 = t_0 + \Delta$, and adapt parameter estimates as needed.

6. Goto step 2.

In order to accommodate the computational burden with very fast dynamics, the general idea of most of the current approaches consists in: (i) solving the optimization problems off-line for the whole state-space using efficient optimization solvers (such as, SQP, multi-parametric programming), leading to Value Functions or to parameterized sets of controllers, with possibly approximating, control laws which are stored in a look up table; and (ii) recruiting pre-stored controllers or extract values from the look-up table to parameterize controllers adapted to the current situation on a real time basis.

While sharing the idea of pre-computing off-line the most computationally demanding building blocks of the MPC scheme with previous work concerning MPC schemes with very fast dynamics, (79), the approach proposed here clearly departs in a very substantive way from the above reported work. This will be clear in the next section.

Before pursuing describing the Attainable Set MPC scheme, let us state the standing assumptions on the data of the problem (P_T) that we will consider form now onwards.

Assumption 1 *The function $g : \mathbb{R}^n \to \mathbb{R}$ is bounded from below, Lipschitz continuous with constant K_g and also C_1. The last two properties imply that the gradient of g is well defined and bounded by K_g everywhere.*

Assumption 2 *The function* $f_0 : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ *is bounded from below, Lipschitz continuous with constant* K_{f_0} *and also* C_1 *in* x, $\forall (t,u) \in [0,\infty) \times \mathbb{R}^m$ *and continuous in* (t,u) $\forall x \in \mathbb{R}^n$. *Thus, the gradient of* f_0 *with respect to* x *is well defined and bounded by* K_{f_0} *everywhere.*

Assumption 3 *The function* $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ *is bounded from below, Lipschitz continuous with constant* K_f *and also* C_1 *in* x, $\forall (t,u) \in [0,\infty) \times \mathbb{R}^m$ *and continuous in* (t,u) $\forall x \in \mathbb{R}^n$. *Thus, the Jacobian of* f *with respect to* x *is well defined and bounded by* K_f *everywhere.*

Assumption 4 *The function* $h : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^q$ *is bounded from below, Lipschitz continuous with constant* K_h *and also* C_1 *in* x, $\forall t \in [0,\infty)$ *and continuous in* t $\forall x \in \mathbb{R}^n$. *Thus, the Jacobian of* h *with respect to* x *is well defined and bounded by* K_h *everywhere.*

Assumption 5 *The sets* $C_f \subset \mathbb{R}^n$ *and* $\Omega \subset \mathbb{R}^m$ *are compact.*

Assumption 6 *The control* $u : [0,\infty) \to \mathbb{R}^m$ *belongs to the set* $\mathcal{U} = \{u \in L^\infty : u(t) \in \Omega,\ \forall t,$ *and* $\forall \mathcal{T} \subset [0,\infty)$ *of finite measure,* $\int_{\mathcal{T}} u(s)ds < \infty$.

Assumption 7 *Finite time controllability: For any* x_1 *and* x_2 *in* $\mathbb{R}^n$, *there exists and interval* $[t_1, t_2]$ *sufficiently large and a control function* $u : [t_1, t_2] \to \Omega$ *steering the system from* x_1 *to* x_2.

Observation. This last assumption, with the help of at last part of the assumptions 1-6, implies the existence of at least one infinite horizon optimal control strategy for the (OCP) converging uniformly to a given $\xi^* \in C_f$.

To see this, let $x(0) = x_0$ and an increasingly monotonic sequence $\{t_i\}_1^\infty$ be a sequence of times such that $t_1 = 0$ and $\lim\limits_{i\to\infty} t_i = \infty$ satisfying assumption 7 and, as such, that $\lim\limits_{i\to\infty} x(t_i) = \xi^* \in C_f$.

Under the above assumptions, it is guaranteed the existence of an unique optimal solution x_i^* to the free-time (OCP) on $[t_i, t_{i+1}]$ such that $x_{i+1}^*(t_{i+1}) = x_i^*(t_{i+1})$ by concatenating the sequence of segments of trajectories $\{x_i^*\}$ we obtain that there exists a feasible optimal control process (x^*, u^*) for the infinite horizon (OCP) such that $\lim\limits_{t\to\infty} x^*(t) = \xi^*$.

These assumptions complemented with some additional more technical requirements will ensure the required properties of the novel MPC schemes proposed in this chapter.

5.3 Attainable Set MPC

5.3.1 Introduction and Preliminaries

In this section, we formulate the Attainable Set MPC (AS-MPC) scheme in the context of a long (possibly infinite) time horizon optimization problem by a sequence of sliding shorter time horizon sub-problems initialized with the current sampled state. Like in the conventional MPC, the control loop is closed by sampling the state variable in order to compensate for the effect of perturbations in the evolution of the state trajectory. Furthermore, this scheme enables the incorporation of features of the environment - e.g., static or dynamic obstacles detected within the appropriate sensors range - which, in spite of being quite natural in many application scenarios such as those involving autonomous vehicles, are not, in general incorporated in the conventional optimal control formulations, and, thus, in the usual associated MPC schemes.

The key idea of the novel MPC Scheme proposed in this work consists in, at each iteration, replacing the optimal control problem to be computed on-line for the prediction horizon, by an equivalent finite-dimensional optimization one which consists in minimizing a certain cost functional on a certain set of the state space. The term "equivalent" here is in the sense that the solution to the new optimization problem is the value of the optimal state trajectory for problem (P) in Chapter 3 at the final time of the current control horizon, i.e., $t_0 + \Delta$ where t_0 is the current time.

For the sake of convenience, we restate the optimal control problem here.

First, we consider the optimization of a dynamic control system over a very long time horizon $[t_i, t_f]$. Remark that, with an appropriate change in the definition of solution concept and assumptions on the data of the problem, this problem can be stated for infinite horizon.

$$\begin{aligned} (P_{t_0}) \text{ Minimize} \quad & g_0(x(t_f)) + \int_{t_0}^{t_f} f_0(t, x(t), u(t))dt \\ \text{subject to} \quad & \dot{x}(t) = f(t, x(t), u(t)), \quad \mathcal{L} - a.e. \\ & x(t_f) \in C_f,\ x(t_0) \text{ is given, with } t_0 \geq t_i \\ & u \in \mathcal{U}, \end{aligned}$$

where $C_f \subset I\!R^n$, and $\mathcal{U} := \{u : [t_i, t_f] \to I\!R^m : u(t) \in \Omega\}$, with $\Omega \subset I\!R^m$ being some closed set. Note that, for infinite horizon, we impose assumptions ensuring the existence of trajectories converging to some equilibrium points in a specified set of the state space.

Before pursuing, let us note that, by enabling various choices of t_f, the above formulation of (P_{t_0}) encompasses various types of MPC schemes. It may be either infinite or finite, and, in the later case, take on a very large value, usually called the prediction or optimization horizons. It can also be considered a moving horizon value, i.e., $t_f = t_0 + T$, where T is the optimization horizon usually considered in the conventional receding horizon MPC scheme. In this scheme, the computational complexity of solving the (OCP) on-line dictates an upper bound on the value of T. As we will see, this issue disappears in the AS-MPC scheme proposed here. However, this observation is of interest since it allows to relate both schemes. This relation will make derivation of AS-MPC properties easier.

The problem (P_{t_0}) of the conventional MPC scheme stated in the previous section is replaced by the following optimization problem

$$
\begin{aligned}
(\bar{P}^{\Delta}_{t_0})\ \text{Minimize} \quad & V(t_0 + \Delta, z) \\
\text{subject to} \quad & z \in \mathcal{A}(t_0 + \Delta; t_0, \bar{x}(t_0))
\end{aligned}
$$

where, t_0, $\bar{x}(t_0)$ and Δ, are, as before, respectively, the current time, the value of the state variable sampled at t_0, and the control horizon duration. Notice that, in the absence of any disturbances or uncertainties we have that $\bar{x}(t_0) = x^*(t_0)$ where $x^*(t_0)$ is the optimal solution to the problem $(\bar{P}^{\Delta}_{t_0-\Delta})$.

It will be clear from the definitions below that problems (P) and $(\bar{P})$ are equivalent if and only if the function $V : I\!\!R \times I\!\!R^n \to I\!\!R$ and the set $\mathcal{A}(t; s, z) \subset I\!\!R^n$ with $t \geq s$ are, respectively, the Value Function of problem (P) and the Attainable Set of the dynamic control system at time t from the point $(s, z = x(s))$. We proceed with these definitions. We consider the optimal control problem (P) stated in Chapter 3 without state constraints and the mixed constraints in order to facilitate the exposition. Remark that there is no loss of generality in the definitions. Obviously, the presence of these constraints entail an increased complexity in the conditions characterizing these objects with the consequent complexity in the associated computational procedures. However, this is not a real issue because it matters only at the off-line stage.

Definition 5.1 The Value Function of problem (P) at point (t, z), with $t \in [t_i, t_f]$, and $z \in I\!R^n$ is given by

$$\begin{aligned} V(t,z) \quad := \min_{u \in \mathcal{U}, \xi \in C_f} \{ \, & g_0(\xi) + \int_t^{t_f} f_0(\tau, x(\tau), u(\tau)) d\tau : \; x(t_f) = \xi, \\ & x(t) = z, \; \dot{x}(\tau) = f(\tau, x(\tau), u(\tau)), \; \mathcal{L}\text{-a.e. on } [t, t_f] \, \} \end{aligned}$$

We will denote by $V_T(t, z)$ - and call it the receding T-horizon Value Function - the Value Function above restricted to the interval $[t, t+T]$ for which $t + T < t_f$ and $\xi \in C_{T,f}$ where $C_{T,f}$ is the Forward Attainable Set at time $t + T$ of the set C_f at time t_f, and whose definition will be given below.

Obviously, the computation of the Value Function by using the definition directly entails a huge computational complexity. However, there is a large class of systems - the so-called positional systems, (112, 113, 114) - for which the Value Function can be obtained by solving the Hamilton-Jacobi-Bellman (HJB) equation, (115). By positional systems, it is meant the pair cost functional and controlled dynamics, for which the optimal control strategy for any interval $[t, T]$ with $T > t$, depends only on the specified initial state trajectory value $z = x(t)$.

The HJB equation is given by

$$\begin{cases} \dfrac{\partial}{\partial t} V(t,x) + \min\limits_{u \in \Omega} \left\{ \left\langle \dfrac{\partial}{\partial x} V(t,x), f(t,x,u) \right\rangle + f_0(t,x,u) \right\} = 0 \\ V(t_f, x(t_f)) = g_0(x(t_f)) \end{cases}$$

In general, the Value Function is, at most, merely continuous, and, thus, the partial derivatives have to be understood in a generalized sense, and the solution concept has to be cast in a nonsmooth context. The appropriate solution concept depends on the properties of the solution which, in turn, depends on the structure of the problem. The solution concepts most used in the literature are in a viscosity, generalized, and proximal normal senses, for, respectively, continuous, Lipschitz continuous, and lower semi-continuous solutions. We will not consider these concepts here to avoid breaking the flow of ideas. For details, one may consult the references (115, 116, 117).

It is important to note that there are a number of results characterizing the interplay between level sets of the Value Function and the forward and backward Attainable Sets of the associated dynamic control system, (118, 119).

There are a number of software packages to solve the HJB equation numerically and thus compute a certain approximation to the Value Function, see, for example, (120, 121, 122, 123). The computational complexity of this equation is huge. However, once computed for time invariant dynamic optimization problems, the approximation to the Value Function is stored in a look-up table and invoked to determine the next optimal control at any point (t, x) in time and phase space. Of course, the Value Function will have to be updated whenever there are changes in the environment or in the system that affects the formulation of the underlying optimal control problem as it follows from the general requirements discussed above.

In practice, this generally requires numerical techniques for a discrete approximation to the continuum time system leading to a following recurrence relation analog to the HJB equation, known also as Bellman equation, which can be solved by dynamic programming optimization method developed by R. Bellman, (124).

Once again, let us consider the dynamic control system of (P_{t_0}).

Definition 5.2 Forward and Backward Attainable Sets.

The Forward Attainable Set at time t, often designated only by Attainable Set, from the state x_0 and time $t_0 \leq t$, (118, 125, 126), is define by

$$\mathcal{A}_f(t; t_0, x_0) := \{x(t) : \dot{x}(\tau) = f(\tau, x(\tau, u) \;\; \mathcal{L} - a.e.,\; u \in \mathcal{U},\; x(t_0) = x_0\}.$$

The set $\mathcal{A}_f(t; t_0, x_0) \subset I\!R^n$ is the set of all points that can be reached or attained at time t with all feasible controls from the initial state $x(t_0) = x_0$. It is important to remark that this definition is extended in straightforward way for the case in which state, mixed or other type of constraints are considered. It suffices to ensure that the control processes to be taken into account satisfy all constraints.

Naturally, the Attainable Set from a given initial set is given by

$$\mathcal{A}_f(t; t_0, C_0) = \bigcup_{x_0 \in C_0} \mathcal{A}_f(t; t_0, x_0).$$

For some $t \leq t_1$ and set $C_1 \subset I\!R^n$, the Backward Attainable Set at time t from the set C_1 at time t_1 is the set of points in $I\!R^n$ from which it is possible to steer the state of the system in the interval $[t, t_1]$ to some point in the set C_1. In other words,

$$\mathcal{A}_b(t; t_1, C_1) = \{z \in I\!R^n : \mathcal{A}_f(t_1; t, z) \cap C_1 \neq \emptyset\}.$$

In the (118, 126), these sets are designated by Reachable or by Reach Sets. However, as it is pointed out in (117, 127), this designation is more appropriate for the set of all points that can be reached for all instants within the time interval. In these works, a characterization of the Reachable Sets is given as the level sets of a certain nonsmooth Value Function.

The computational complexity associated with the computation of both types of Attainable Sets is, from their definitions, obviously huge. It is not surprise that a lot of research effort has been put in finding efficient ways of approximating these sets. This will be addressed in the next section.

5.3.2 Formulation of the Attainable Set MPC

As stated before, the fundamental computational burden of conventional MPC schemes is the on-line solving of the optimal control problem in the chosen receding horizon. Thus, the key novelty of the AS-MPC is precisely an on-line computationally very light reformulation of this optimal control problem. This is the first item of this subsection.

Let t_0 be the current time and $\Delta > 0$ be such that $t_0 + \Delta < t_f$. Then, the Principle of Optimality, together with the definition of Value Function on the interval $[t_0 + \Delta, t_f]$, entails that (P_{t_0}) is equivalent to the following finite horizon optimal control problem

$$\begin{aligned} (P_{t_0}^{\Delta}) \text{ Minimize} \quad & V(t_0 + \Delta, x(t_0 + \Delta)) + \int_{t_0}^{t_0+\Delta} f_0(\tau, x(\tau), u(\tau)) d\tau \\ \text{subject to} \quad & \dot{x}(\tau) = f(\tau, x(\tau), u(\tau)), \quad \mathcal{L}\text{-a.e. on } [t_0, t_0 + \Delta] \qquad (5.1) \\ & u \in \mathcal{U}, \text{ and } x(t_0) \text{ is given.} \qquad (5.2) \end{aligned}$$

In order to cast the above optimal control problem in the form of $(\bar{P}_{t_0}^{\Delta})$, we need to express the dynamic constraints in terms of the Attainable Set. However, since we have a running cost, we need first to perform a straightforward change of variable. Let $\tilde{x} = (x, y)$ where

$$\begin{cases} \dot{y} = f_0(t, x, u) \\ y(t_0) = 0, \end{cases}$$

and $\tilde{V}(t, \tilde{x}) = V(t, x) + y$.

By using the definition of Forward Attainable Set in the context of $\tilde{x} = (x, y)$, we obtain

$$(\bar{P}_{t_0}^{\Delta}) \text{ Minimize } \quad \tilde{V}(t_0+\Delta, z)$$
$$\text{subject to } \quad z \in \tilde{\mathcal{A}}_f(t_0+\Delta; t_0, \tilde{x}(t_0)).$$

If $\tilde{x}$ is the reference optimal trajectory, then the solution z^* to $(\bar{P}_{t_0}^{\Delta})$ is given by $z^* = \tilde{x}(t_0+\Delta)$. From now on, without any risk of confusion, we dispense with relabelling and consider $x = \tilde{x}$, $V = \tilde{V}$, and $\mathcal{A} = \tilde{\mathcal{A}}$.

There is an important remark here: Since t_f is, in practical applications, very large - in fact, $t_f = \infty$ is often considered -, we have that $t_f > t_0 + T$, where T is the optimization or prediction horizon considered in the conventional scheme, the scheme proposed here yields, in the absence of any perturbations, the true optimum over the whole running time horizon. This is not the case of the conventional scheme which, by using the optimal control problem (P_T), just yields an approximation of the true global-horizon optimum, which depends on how large T is. Obviously, by choosing $t_f = t_0 + T$, and V_T instead of V in the formulation of the AS-MPC, we have the equivalent to the conventional receding horizon MPC scheme introduced earlier in this chapter.

Now, we are ready to formulate the basic AS-MPC scheme. Clearly, the original infinite dimensional optimization problem was formally expressed by an equivalent finite dimensional one. The complexity was transferred from the control variable to the cost functional - the Value Function - and the set constraints - the Forward Attainable Set - of the optimization problem.

Thus the MPC scheme using this formulation requires (i) the update of the Attainable Set at the end of the current control horizon starting on the current value of the sampled state variable, and (ii) the propagation of the cost functional V over (t, x)-space in order to ensure consistency. Notice that this formulation of the optimization problem exhibits extremely important advantages inherent to its intrinsically geometric character, namely in what concerns the incorporation of additional constraints as well as uncertainties in the dynamics. In particular, this facilitates the consideration of intricate dynamics or partially known environments. Remark that if a time invariant scenario - system and its environment - is considered, the on-line computational burden is minimal: while (i) only requires translations and rotations of the stored onboard

Attainable Set, (ii) involves retrieving the values of the Value Function in the region $(t, x) \subset I\!R \times I\!R^n$ associated with the update of the Attainable Set in (i).

The basic the AS-MPC scheme can be formulated as follows.

Let Δ be the control horizon, and t_0 the current time. Then, the AS-MPC scheme can be formulated as follows:

1. Initialization: $t = t_0$, $x(t_0)$
2. Solve $(\bar{P}^{\Delta}_{t_0})$ to obtain z^* and compute or retrieve (this will be explained later in the chapter when dealing with Attainable Set approximation) the control $u^*_{|[t_0, t_0+\Delta]}$ steering the state variable from $\bar{x}(t_0)$ to z^*, where $\bar{x}(t_0)$ is the sampled state variable at t_0. In case of need, the specific method to compute u^* can be of a direct type which depends strongly on the considered dynamics. However, a general method, to which one can always resort is the PMP.
3. Apply u^* during $[t_0, t_0 + \Delta]$
4. Sample x at $t_0 + \Delta$ to obtain $\bar{x} = x(t_0 + \Delta)$
5. Slide time by Δ, i.e., $t_0 = t_0 + \Delta$, and goto 2.

Remark that, if the goal is to maximize the overall performance (i.e., the total time interval), than this scheme necessarily yields better performance that the conventional T-receding horizon conventional MPC scheme. Obviously, this same scheme in which the cost functional of $(\bar{P}^{\Delta}_{t_0})$ is V_T instead of V, then, performance-wise, it is equivalent to the standard T-receding horizon conventional MPC scheme.

5.4 Properties of the AS-MPC scheme

In this section, we will focus some of the major properties for MPC schemes in the context os the proposed AS-MPC scheme: optimality, asymptotic stability, and robustness. Before pursuing with this agenda, let us note that it is relevant to specify whether we are considering V or V_T as defined in the previous sections, or whether t_f is finite or infinite.

While asymptotic stability only makes sense for $t_f = \infty$, both finite and infinite t_f can be considered for the other two properties. On the other hand, in the absence of perturbations or uncertainties, for either finite or infinite t_f, it is readily concluded that the AS-MPC with the (OCP) formulated using V, it is obvious that the feedback control strategy generated by the AS-MPC yields the global optimum. In this case, it remains of interest to examine the case for which V_T is used. As to the robustness property, we will discuss thee main approaches and justify the one adopted in the next section: update the applied control with intermediate control-horizon state feedback. Here, we also discuss how existing results in the literature for the several ways in which robustness can be considered can easily migrate to the AS-MPC context.

5.4.1 Optimality

As stated above we consider V_T instead of V in the (OCP) associated with the AS-MPC scheme, that is, the Value Function is computed by considering the time horizon $[t_0, t_0 + T]$ where T_0 is the current time and T is the prediction horizon. Let Δ, $\Delta < t$ be the control horizon, and denote by $(x^*_{T,\Delta}, u^*_{T,\Delta})$ the MPC optimal control process obtained with these prediction and control horizons. Denote by $J(x, u)$ the value of the cost functional associated with the control process (x, u) for the optimal control problem set above with V_T expressed in the Lagrange form over[1] $[0, \infty)$, by $J(x, u)_{|[\alpha,\beta]}$ be its restriction to the interval $[\alpha, \beta]$, and by $J_k(x, u)$ a short notation for the case with $\alpha = k\Delta$ and $\beta = (k + 1)\Delta$.

Theorem 5.4.1 *Let $t_0 = 0$, and assume that the optimal control horizon has an optimal control process (x^*, u^*) such that $\lim_{t\to\infty} x^*(t) = \xi^*$, being ξ^* an equilibrium point in C_∞, the state final point constraint set. Moreover, assume that there are no perturbations and no uncertainties. Then,*

i) $\displaystyle\lim_{\Delta\downarrow 0,\, T\uparrow\infty} \sum_{k=0}^{\infty} J_k(x^*_{T,\Delta}, u^*_{T,\Delta}) = J(x^*, u^*)$, *and*

ii) Consider the (OCP) of the AS-MPC the with cost specified by V instead of V_T as in i). Then, $\displaystyle\lim_{\Delta\downarrow 0,\, k\uparrow\infty} \left| J_k(x^*_\Delta, u^*_\Delta) - J(x^*, u^*)_{|[k\Delta,(k+1)\Delta]} \right| = 0.$

Remark that item *i)* also holds for the case of a finite t_f with $T \uparrow \infty$ replaced by $T \uparrow t_f$, and $\Delta \downarrow 0$ in such a way $\lim_{k\to\infty} k\Delta = t_f$.

[1] Given an interval $[t_0, t_1]$, the Lagrange form of a C_1 cost functional ϕ that depends on the state variable at the final time t_1 is given by $\phi(t_0) + \int_{t_0}^{t_1} \nabla\phi(x(t) f(t, x(t), u(t)) dt$

This proof will be done for the case without state constraints. This does not bring any loss of generality since, under our assumptions, it is not difficult to see that, by using a standard penalization technique, an equivalent Optimal Control Problem (OCP) without state constraints can be formulated. We skip this step.

Proof Let us start with statement i). Fix Δ and T with $\Delta < T$. It follows straightforwardly from the principle of optimality that a globally optimal control process $(x^*_{T,\Delta}, u^*_{T,\Delta})$ is the concatenation of segment-wise optimal control processes if, for any segment, the final value of the state variable at that segment is equal to the initial value of the optimal trajectory of the ensuing segment. Denote by $\xi^*_{T,\Delta}$, the final value of $x^*_{T,\Delta}$.

Since $\lim\limits_{t\to\infty} x^*(t) = \xi^*$, $\exists\, T_\varepsilon$ sufficiently large such that $x^*_{T_\varepsilon}$, the solution to the (OCP) restricted to $[0, T_\varepsilon]$ satisfies $x^*_{T_\varepsilon}(T_\varepsilon) \in x^*(T_\varepsilon) + \varepsilon B$.

Now, for $k = 0, \ldots$, let $\delta > 0$ be such that the solution to the (OCP) in the tube

$$\mathcal{T}^{T_\varepsilon}_\delta(x^*, u^*) := \{(x, u) : x \in x^*(t) + \delta B_{R^n},\ u(t) \in (u^*(t) + \delta B_{L_\infty}) \cap \mathcal{U},\ \forall t \le T_\varepsilon\}$$

is unique for each initial value $x^*_{T_\varepsilon}(T_\varepsilon)$ and thus, by considering a simple transformation whereby the cost functional of the (OCP) depends only on the state variable at the final time, we have that, within the tube $\mathcal{T}^{T_\varepsilon}_\delta$, $\lim\limits_{t\to\infty} x^*_{T_\varepsilon}(t) = \xi^*$.

Then, for any given $\Delta > 0$, and for $k = 1, 2, \ldots$, the same argument holds by shifted time interval $[k\Delta, T_\varepsilon + k\Delta]$. We conclude that $\lim\limits_{t\to\infty} x^*_{T_\varepsilon + k\Delta}(t) = \xi^*$. Now, for any given Δ, let us consider k such that sufficiently large so that $k\Delta > T_\varepsilon$, we easily reach the conclusion that

$$\lim_{\bar{k}\to\infty} (x^*_{T_\varepsilon,k}, u^*_{T_\varepsilon,k})_{[\bar{k}\Delta,(\bar{k}+1)\Delta]} = (x^*, u^*)_{[\bar{k}\Delta,(\bar{k}+1)\Delta]}$$

in the norm $AC([0,\infty)) \times L_\infty$.

Thus, for any $M > 0$, $\lim\limits_{T_\varepsilon\to\infty,\Delta\to 0} \sum\limits_{k=0}^{M} J_k(x^*_{T_\varepsilon,k}, u^*_{T_\varepsilon,k}) = \sum\limits_{k=0}^{M} J_k(x^*, u^*)$. Finally, and by shortening the notation, we conclude that

$$\lim_{M\to\infty} \sum_{k=0}^{M} J^*_k = J(x^*, u^*).$$

A simple contradiction argument reveals that (ii) follows immediately from (i) for any fixed Δ.

5.4.2 Asymptotic Stability

In this subsection we show that the proposed AS-MPC scheme generates asymptotically stable control strategies under some reasonable sets of assumptions. We will consider two different contexts: a more classical on that requires the inclusion of a stabilizing finite state set constraint at the finite time and another one that dispenses with it.

Let us start with the first one that draws heavily from Theorem of (128). In this approach, we consider without any loss of generality - since if $f(x_s, u_s) = 0$, one can always shift the origin of the system to (x_s, u_s) - that $f(0,0) = 0 \in \mathbb{R}^n$ and, thus $0 \in \mathbb{R}^n$ is an equilibrium at infinity with $u = 0$. Moreover, let us consider that the data of our problem satisfies the following - somewhat mild - assumptions:

(A) $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is twice continuously differentiable.

(B) $\Omega \subset \mathbb{R}^m$ is compact, and convex, and 0 is in the interior of Ω.

(C) The dynamic control system has a unique solution for any initial condition $x_0 \in R^n$ and any piecewise continuous and right-continuous $u : [0, \infty) \to \Omega$.

It is not difficult that these assumptions can be easily weakened. However, this would entail a more cumbersome presentation of the arguments.

Essentially, we show that, under these assumptions as well as within the mild context considered in (128), that, in the absence of disturbances, the optimal control process generated by the AS-MPC applied to the receding horizon $[t_0, t_0 + T]$ satisfies the same requirements as the ones generated by the MPC scheme in (128), and thus its main result – that we state below for convenience – can be applied. This result can be stated as follows:

Theorem 5.4.2 *Let the following assumptions hold.*

1) Assumptions (A)-(C) are satisfied,

2) The Jacobian linearization of the given nonlinear dynamic system is stabilizable,

3) The open-loop optimal control problem underlying the MPC scheme feasible on $[t_0, \infty)$.

Then, for a sufficiently small sampling time Δ and in the absence of disturbances, the PC closed-loop system is asymptotically stable.

If we denote by V_T, the Value Function defined on the horizon $[t_0, t_0+T]$ for a given finite T, then it is clear that the AS-MPC with an (OCP) with V_T as cost functional is equivalent to the conventional MPC scheme with stabilizing final state constraints. Thus, by imposing the additional constraints required in this result, the above Theorem can be applied to our context and yield the asymptotic stability of the control processes generated by AS-MPC scheme with V_T replacing V,

The above result is a classic one and requires assumptions on the data of the problem that are relatively strong is spite of the range of its applicability being very significant. However, it exhibits the need of imposing a final state constraint set with the sole technical purpose of ensuring the asymptotic stability. This assumption might be too taxing and more recent results have shown that if the cost functional contains a term that depends on the state variable at the final time, then, it is possible to show the asymptotic stability without requiring this final constraint set technical requirement, (129).

One popular way to avoid the technical final set constraint, is to assume a local controllability property or, almost equivalently, to ensure asymptotic (in the sense of taking the limit in T) converging bounds on the Value Function, in order to show that every level set of the infinite horizon optimal Value Function is contained in the basin of attraction of the asymptotically stable equilibrium for sufficiently large optimization horizon T.

In chapter 6 of (129) a comprehensive stability and sub-optimality analysis for MPC schemes for autonomous nonlinear systems without stabilizing terminal constraints are presented, in the discrete-time context. The important advantage in using the asymptotic controllability assumption on the (OCP) for which it is possible to derive detailed asymptotic stability and performance estimates. Together with the first three assumptions considered in the previous asymptotic stability, the controllability assumption enables to derive estimates on the level of sub-optimality and bounds on the optimization horizon which play a critical role in ensuring stability with parameters explicitly computed from the controllability condition. These results can easily migrate to the AS-MPC, in which the controllability condition for the (OCP) takes the form (note that to facilitate the arguments and the notation, we consider, without any loss of generality, time invariant systems):

Controllability assumption. The system is asymptotically controllable with respect to f_0 with rate $\beta \in \mathcal{KL}_0$, if and only if, for each $x \in I\!R^n$ and each $T > 0$ there exists an admissible control sequence $u_x \in \mathcal{U}_T(x)$ satisfying

$$f_0(x(t,x_0),u(t)) \leq \beta(f_0^*(x),t), \;\; \forall\, t \in [0,T].$$

Here, $\mathcal{KL}_0$ is the class of functions $\beta : I\!R^+ \times I\!R^+ \to I\!R^+$ that satisfies $\beta(0,\tau) = 0$ and $\lim_{r\to\infty} \beta(r,\tau) = \infty$ and $\forall r > 0$, $\lim_{\tau\to 0} \beta(r,\tau) = 0$, and $f_0^*(x) = \min_{u\in\Omega}\{f_0(x,u)\}$.

Following the same arguments of in chapter 6 of (129), when the above assumptions hold, we have the estimates:

- For any $T > 0$, and $x \in I\!R^n$, we have $V_T(x) \leq J_T(x,u_x) \leq \displaystyle\int_0^T \beta(f_0^*(x),t)dt$, for all u_x satisfying the controllability assumption.

- Let $x_0 \in I\!R^n$ and u_T^* solution to (OCP) restricted to $[0,T]$. Then, for any $\bar{T} \in (0,T)$, $V_T(x_{u^*}(\Delta,x_0) \leq J_{\bar{T}}(x_{u^*}(\Delta,x_0),u^*(t)) + \int_0^{T-\bar{T}} \beta(f_0^*(x_{u^*}(\Delta+\bar{T},x_0)),t)dt$, for $t \geq \Delta$.

- Take u_T^* as in the previous item and let $N = T/\Delta$. Then,, for $k = 0,1,\ldots N-1$, we have that $J_{N-k}(x_{u^*}(k\Delta,x_0),u^*(k\Delta+t)) \leq \int_0^{T-k\Delta} \beta(f_0^*(x_{u^*}(k\Delta,x_0),t)dt$, $\forall t > 0$.

In the above and in what follows x_u, and $x_u(t,\tau,x_\tau)$ with $x(\tau) = x_tau$ and $\tau \leq t$, represent, respectively, the state trajectory at time t associated with the feasible control u, independently of the initial state, and the state trajectory associated with the feasible control u and time $t \geq \tau$ with the trajectory initiated at $x(\tau) = x_\tau$. In this last case the middle argument may be omitted if it is obvious from the context.

From these estimates, and by using elaborated arguments one concludes the time-continuum equivalent to Theorem 6.18 of (129) which establishes the asymptotic stability of the MPC scheme with more precise estimates than the ones considered in the previous stability result. Like before the relation between the standard MPC and the variant of the AS-MPC with V_T in place of V, validates this result for our context. The fact that $V < V_T$ implies that it also holds for our AS-MPC scheme.

Similar results with arguments with somewhat different flavor are discussed in (72, 130).

5.4.3 Robustness

The above properties were obtained by assuming the absence of disturbances and uncertainties. However, the fragility - in the sense of lack of robustness - of the conventional MPC schemes is well known due to the fact that they rely strongly on complex optimization processes. Thus, it is clear that robustness is a primary concern when applying MPC to real-life problems. It is clear that un-modelled interferences, measurement and input noises, as well as the associated quantization errors due to limitations in both computer memory and computational times interfere in the behavior of the system with, possibly, very significant impact in the optimization process.

So it is of interest to obtain relations between bounds of the disturbances and uncertainties with some measure of the extent to which the properties listed above are preserved. The notion of robustness comprehends many scopes, notably, in what concerns constraint satisfaction, stability, optimized performance, and computational practicability.

A cursory overview reveals that the integrated complexity of the issues underlying robustness is far from being satisfactorily addressed in that the convoluted interplay between the quality of the guaranteed structural properties and the complexity of associated computational methods still require a lot of research. Most of the literature concerns paradigms that concern partial aspects of robustness, such as min-max open loop, min-max closed loop, trajectory tube formulation, coupling receding horizon control (RH) with RH estimation, to just name the most significant ones.

One of the most popular general approaches to ensure the asymptotic stability robustness consists in considering assumptions on the data of the problem, and the MPC scheme formulation that ensure the continuity of the Value Function, and, moreover, in an uniform neighborhood along the reference trajectory that contains feasible controls in which it also satisfies a Lyapunov inequality in a generalized sense, (73). It is clear that the applicability of this approach is limited, mainly because of the simple fact that a great advantage of the (OCP) is precisely the consideration of a wide range of constraints. However, these features are a critical obstacle to the success of the approach.

The replacement of the (OCP) in the MPC scheme by a minimax is another way o mitigate the effect of disturbances, (72). However, by considering the worst case of the perturbations, this approach typically leads to very conservative performances.

Another approach that became quite popular consists in considering tubes of trajectories instead of a single reference trajectory, (71, 73). The success of this approach is very much related to striking the best trade-off between conservatism of the solution and the computational complexity involved. The fact that, in spite of being around for some time, these results have not yet been proved themselves in the real-world applications, keeps the expectations open concerning future developments of this approach.

Finally, the so-called multi-steps (in (77)) or intermediate steps (in (7)) emerged in recent years. The idea is to close the loop at intermediate points in the time interval of the receding-horizon optimization. These two approaches are different but there are many common issues in the robustness analysis. The approach in (7) will be further developed in a later section of this chapter.

5.5 Attainable Set Approximation Approaches

Although this problem exhibits a much simpler appearance, the fact is that both the Attainable Set and the Value Function are extremely complex objects whose computation is of a very high complexity, usually comparable to that of solving the corresponding HJB equation. Thus, it is not surprising that a number of approaches have emerged to approximate Attainable Sets. This will be the subject of the this section.

In this section, we will present a quick comparative survey of the three most significant approaches to compute a set approximating a given set: Ellipsoidal, Polyhedral, and ε-Dense Discrete Set, often referred to by “Cloud of Points”. For first two, a brief overview of the literature will be given. In what concerns the third, this is to best of our knowledge the first time that this type of approximation to sets is being considered, at least, in the control context. Then, the reasons why the selection of the last approach for the implementation of the AS-MPC will be given.

5.5.1 Overview

- Ellipsoidal Approximations. There has been a large number works developing methods to compute approximations to Attainable Sets by ellipsoidal sets. Among many others, landmarks articles are (131, 132).

These works concern the computation of approximations to the Attainable Sets for discrete or continuous time linear control systems with time-varying coefficients and ellipsoidal bounds on the controls and initial conditions in a first stage, and, later, were extended to nonlinear dynamic systems that could be easily linearizable and, thus, faithfully approximated by piecewise linear systems. Algorithms generating either external or internal ellipsoidal approximations that touch the Attainable Set boundary from outside and from inside were produced. While the former are useful to ensure safety properties, that is, the system does not enter forbidden unsafe region, the later, make sure that there are control strategies whose trajectories reach points in given sets by concluding that their intersection with the inner ellipsoidal approximation is nonempty.

There are great computational advantages of this approach. Ellipsoidal sets are characterized by a small number of parameters which, for linear or piecewise linear systems can be easily propagated by solving ordinary differential equations with coefficients given in explicit analytical form. This allows exact parametric representation of reach tubes through families of external and internal ellipsoidal tubes as compared with earlier methods based on constructing one or several isolated approximating tubes. This approach gave rise to efficient numerical algorithms.

However, there are a couple of fundamental drawbacks associated with this approach. The most important one concerns the fact that the approximations - either internal or external - are usually too conservative, that is, they lead to control strategies that, albeit satisfying the imposed requirements, they might not be the ones more appropriate to achieve appropriate trade-offs with other criteria such us performance, satisfaction of constraints, robustness, safety, etc.

- Polyhedral Approximations. This approach has been widely considered in the literature, (125, 133, 133, 134, 135), among others. Part of the work of the last reference is expanded in the annex "Polyhedral Set Approximation" of this thesis. It consists essentially in finding the vertices of a polyhedron which lie on the boundary of the Attainable Set, and, then, taking their convex hull. Constructions of inner and outer approximations have been developed. The main work along this line took place for linear systems, but extensions for significant classes of non linear systems were also developed. The essential idea is to regard a vertex of the polyhedron approximating the Attainable Set as the final value of the optimal trajectory to an Auxiliary Optimal Control Problem (AOCP) in which a cost functional linear in the state variable at the final time is minimized subject to the dynamics and other constraints of the given dynamic control system. The solution to this AOCP gives a boundary point of the Attainable Set that depends on the coefficient of the cost functional. By judiciously varying this coefficient, a number of adequate points of the boundary of the Attainable Set is generated. The term "adequate" means that a least conservative inner or outer approximation are obtained. Moreover, schemes to address nonconvex Attainable Sets where also developed (see the corresponding annex).

This method has been the subject of further research and, since it revealed to be not only too specific, but also of limited impact in the sequel of the works of this thesis, a more detailed account of its results and consequent algorithms are included in the annex "Polyhedral Set Approximation".

- ε-Dense Discrete Set. This approach relies strongly and, in some respects, can be considered an application of one the more elegant ways of approximating Attainable Sets, the so called exponential formula introduced by Peter Wolenski in the differential inclusions context in the article (127).

  To present the main result of this article, it is of interest to introduce a number of key ideas: (i) relation between controlled ordinary differential equations and differential inclusions; (ii) composition of se-valued maps; and (iii) Limits of sequence of sets. In this last item, we will restrain ourselves to the limit in the sense of Kuratowski.

  Let us start with (i). For the sake of the presentation simplicity, let us consider the time invariant dynamic system $\dot{x} = f(x, u)$ where the control u is such that $u(t) \in \Omega$. Let us assume that the standing assumptions considered earlier in this chapter are in force. Let us consider the set-valued dynamics $\dot{x} \in F(x)$, where $F : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^n)$ is assumed to Lipschitz continuous in the sense of Hausdorff with respect to x, i.e., $\exists K_F > 0$ such that $d_H(F(x), F(y)) \leq K_F \|x - y\|$, and $\dot{x} \in \{ f(x, u) : u \in \Omega \}$.

Here, and in what follows, the Hausdorff distance between sets A and B, $d_H(A,B)$, is defined by $d_H(A,B) := \max\left\{\sup_{x\in A}\{d_B(x)\}, \sup_{y\in B}\{d_A(y)\}\right\}$, where $d_A(a)$ is the Euclidian distance between the point a and the set A.

Even if, $\forall x \in I\!\!R^n$, $F(x) \equiv \{\, f(x,u) : u \in \Omega\}$, it is clear that, in general, the set of solutions to the differential inclusion $\dot{x} \in F(x)$ is much larger that of the solutions to $\dot{x} \in \{\, f(x,u) : u \in \Omega\}$ as it can be easily seen that the former includes controls in the feedback form. It is a well known result that the two systems are equivalent only if f is also continuous in the control u.

In order to consider (ii), i.e., to define the composition of set-valued map, consider two set-valued maps $F_1 : I\!\!R^n \to \mathcal{P}(I\!\!R^n)$ and $F_2 : I\!\!R^n \to \mathcal{P}(I\!\!R^n)$ with domains and values in a linear vector space. Define the composition $(F_1 \circ F_2)(x) := \{y \in I\!\!R^n : \exists \bar{y} \in F_2(x) \text{ such that } y \in F_1(\bar{y})\}$. If a set-valued map $F(x)$ is composed with itself N times, we denote the resulting set-valued map by $F^N(x)$.

Let $d(a,A)$ be the usual distance of the point a to the set A, i.e., $\inf_{\bar{a}\in A}\{\|a-\bar{a}\|\}$. Given a sequence of sets $\{A_n\}$ in $I\!\!R^n$, let us consider the following notions of limits

$$\begin{aligned}
\liminf_{n\to\infty} A_n &:= \{x \in I\!\!R^n : \limsup_{n\to\infty}\ d(x,A_n) = 0\}\\
\limsup_{n\to\infty} A_n &:= \{x \in I\!\!R^n : \liminf_{n\to\infty}\ d(x,A_n) = 0\}.
\end{aligned}$$

It is said that the sequence $\{A_n\}$ converges to some set $A \subset I\!\!R^n$ in the sense of Kuratowski if and only if $\liminf_{n\to\infty}\ A_n = \limsup_{n\to\infty}\ A_n = A$

Now, we are ready to state the main result in (127) that provides the so called exponential formula for dynamic systems given by differential inclusions.

Assume that the set-valued map F has non-empty, compact, convex values on $I\!\!R^n$ and that is locally Lipschitz on $I\!\!R^n$. Then, for all $x \in I\!\!R^n$ and $\Delta > 0$, we have the Attainable Set of F is given by

$$\mathcal{A}_F(\Delta; 0, x_0) := \lim_{N\to\infty}\left(I + \frac{\Delta}{N}F(x)\right)^N, \tag{5.3}$$

where I is the identity matrix, the set products are in the sense of set-valued maps composition, and the limiting operation is defined in the sense of Kuratowski. This formula is proved in (127) for Lipschitz differential inclusion with convex values but it may be further extended for more general set-valued maps. The proof of the main theorem partially relies on a C^1 approximation result due to Filippov, for which a new proof is given. Moreover, in this paper, this formula is used to derive a characterization of the Value Function associated with an (OCP) with the dynamics given by the considered differential inclusion.

The ε-Dense Discrete Set approximation consists in selecting a positive number ε, a subset of discrete points of the set-valued map $F(x)$, each one corresponding to the velocity of the system with a certain piecewise constant control for $\bar{u}$, and in choosing a certain number N_ε so that, by truncating the limit at $N = N_\varepsilon$, a sufficiently large of points z_i of the Attainable Set points is obtained so that

$$\mathcal{A}(\Delta; 0, x_0) \subset \cup_{i=1}^{N_\varepsilon}[z_i + \varepsilon B_1(0)]$$

and $d_H\left(\mathcal{A}(T; x_0, 0), \left[\cup_{i=1}^{N_\varepsilon}[z_i + \varepsilon B_1(0)]\right]\right) < \varepsilon$. Here, $B_1(0)$ is the closed unit ball centered at the origin.

This type of approximation was selected to the implementation of the AS-MPC scheme since it is endowed with nice properties and, moreover, it is the one that imposes the least on-line computational burden. The constructive procedure based on Wolenski's exponential formula will be detailed in the next section.

5.5.2 The ε-Dense Discrete Set Approximation

In this section, we consider $F(x) = f(x, \Omega)$ where $\Omega \subset \mathbb{R}^m$ is the set of values that the control variable can take. We assume that f is Lipschitz continuous in x and continuous in u. These properties ensure not only the existence and uniqueness of the solution of $\dot{x} = f(x, u)$ for a given feasible control u and initial condition $x(0) = x_0$ but also the equivalence between the differential inclusion and ordinary differential representations in the sense that they have the same set of solutions. We remark that the fact that we are considering now only autonomous systems does not constitute any loss of generality. These developments can be easily transposed for time variant systems.

Consider positive integers N_u, N_Δ sufficiently large, and δ_u adequately small. Define

$$\Omega_{N_u} := \{u_i \in \mathbb{R}^m : u_i \in \Omega,\ i = 1, \ldots, N_u\}$$

satisfying the conditions (i) $\Omega \subset \Omega_{N_u} + \delta_u B_1(0)$, (ii) $\forall i\ \exists j$ s.t. $\|f(x,u_i) - f(x,u_j)\| < \delta_u$ and (iii) $\forall x \in I\!R^n$, $\forall v \in \partial f(x,\Omega)$, $\exists \bar{v} \in \partial f(x,\Omega_{N_u}) \cap \partial f(x,\Omega)$ such that $\|v - \bar{v}\| \leq \delta_u$. Here, ∂A represents the boundary of the given closed set A.

Now, to construct $\mathcal{A}^{N_\Delta}_{F_{N_u}}(\Delta; 0, x_0)$, we just consider the truncation up to N_Δ of the exponential formula which, was defined in the previous section. The integer power of the is understood in the sense of composition of set-valued maps. To illustrate this point, just let $N_\Delta = 2$, then

$$\mathcal{A}^{2}_{F_{N_u}}(\Delta; 0, x_0)(I + \frac{\Delta}{2}F_{N_u}(x_0))^2 = \bigcup_{i=1}^{N_u} \left\{ \left(I + \frac{\Delta}{2}F_{N_u}(x_i) \right) : x_i \in I + \frac{\Delta}{2}F_{N_u}(x_0) \right\}.$$

For this simple example, it is clear that the approximation of the value of the state variable at time Δ is obtained by considering a discrete system in which the controls are composed by two piecewise constant segments, each of duration $\dfrac{\Delta}{2}$. Obviously, the larger the number N_u, the better will be the approximation of the Attainable Set. Another observation consists in the fact that, for each $z_i \in \mathcal{A}^{N_\Delta}_{F_{N_u}}(\Delta; 0, x_0)$ we may associate a piece-wise constant control function with N_Δ segments and taking values in Ω_{N_u}.

Now, we detail the construction of the ε-Dense Discrete Set approximation and discuss, a good estimate of the Hausdorff distance between these sets to determine the worst case of sub-optimality.

Let N_u be a given sufficiently large integer, and Ω_ε denote the set $\{u_i \in \Omega : i = 1, \ldots, N_\varepsilon\}$ satisfying the following properties:

i) $\Omega \subset \displaystyle\bigcup_{i=1}^{N_\varepsilon} (u_i + \varepsilon B)$, and

ii) $\forall i$, $\exists j$ s.t. $\|f(t,x,u_i) - f(t,x,u_j)\| < \varepsilon$.

iii) $\forall x \in I\!R^n$, $\forall v \in \partial f(t,x,\Omega)$, $\exists \bar{v} \in f(t,x,\Omega_{N_u}) \cap \partial f(t,x,\Omega)$ such that $\|v - \bar{v}\| \leq \varepsilon$.

Denote by $\mathcal{A}_f(t_1; t_0, x)$ and $\mathcal{A}^{\varepsilon}_f(t_1; t_0, x)$ the points attainable at $t_1 > t_0$ from x at t_0, by the dynamic system with controls, respectively, in L^∞ with values in Ω, and piecewise constant with values in Ω_ε.

Now, we are ready to state the following property.

Proposition 5.5.1 *Let Δ be a positive number. Under mild assumptions on the dynamics, we have, for any $(t,x) \in I\!R \times I\!R^n$,*

$$d_H\left(\mathcal{A}_f(t+\Delta;t,x), \mathcal{A}_f^{\varepsilon}(t+\Delta;t,x)\right) \le \varepsilon\Delta e^{K_f\Delta}.$$

Proof Fix an arbitrary $\tilde{u} \in \mathcal{U}_{|[t,t+\Delta]}$, and let us consider a piecewise constant $\bar{u} \in \Omega_\varepsilon$. We have that

$$\begin{aligned} x_{\tilde{u}}(t+\Delta;t,x) - x_{\bar{u}}(t+\Delta;t,x) &= \int_t^{t+\Delta} [f(s,\tilde{x}(s),\tilde{u}(s)) - f(s,\bar{x}(s),\bar{u}(s))]ds \\ &= \int_t^{t+\Delta} [f(s,\tilde{x}(s),\tilde{u}(s)) - f(s,\bar{x}(s),\tilde{u}(s))]ds \\ &\quad + \int_t^{t+\Delta} [f(s,\bar{x}(s),\tilde{u}(s)) - f(s,\bar{x}(s),\bar{u}(s))]ds \end{aligned}$$

From the above conditions, it follows that the piecewise constant $\bar{u} \in \Omega_\varepsilon$ can be chosen with a sufficiently large number of points in the partition of $[t, t+\Delta]$ so that have that

$$\| \int_t^{t+\Delta} [f(s,\bar{x}(s),\tilde{u}(s)) - f(s,\bar{x}(s),\bar{u}(s))]ds \| \le \varepsilon\Delta.$$

It follows, by the above and the Lipschitz continuity of f w.r.t x, that we can write

$$\|x_{\tilde{u}}(t+\Delta;x,t) - x_{\bar{u}}(t+\Delta;x,t)\| \le K_f \int_t^{t+\Delta} \|x_{\tilde{u}}(s;x,t) - x_{\bar{u}}(t+s;x,t)\|ds + \varepsilon\Delta.$$

By applying the Bellman-Gronwall inequality, we conclude that $\forall \tilde{u}$ taking values in Ω, $\exists \bar{u}$ piecewise constant control taking values in Ω_ε such that

$$\|x_{\tilde{u}}(t+\Delta;t,x) - x_{\bar{u}}(t+\Delta;t,x)\| \le \varepsilon\Delta e^{K_f\Delta}.$$

However, this means that $\forall z \in \mathcal{A}(t+\Delta;t,x)$, $\exists z_\varepsilon \in \mathcal{A}_\varepsilon(t+\Delta;t,x)$ such that $\|z - z_\varepsilon\| \le \varepsilon\Delta e^{K_f\Delta}$. This is a sufficient condition for the conclusion of the proposition.

Some comments are pertinent to emphasize the relevance of this result:

- The rationale of selecting points with constant controls relies on the fact that, from the computational point of view it is important to approximate the desired trajectory by a sequence of trajectories generated by piecewise controls. For many applications, box-type of control constraints are relevant, and, in this case, the application of the Maximum Principle yields controls which are piecewise constant.

- In what concerns the on-line computational burden of the AS-MPC, it is relevant to generate a grid of points of the Attainable Set with the corresponding piecewise constant control, which can be stored in the on-board look-up table.
- Moreover, there many classes of control systems for which the application of even constant controls during the control horizon Δ of the AS-MPC scheme suffices.

Since the AS-MPC scheme encompasses the optimization of the Value Function over a certain transformation of the Attainable Set from each sampled state, and, as explained earlier, both the Value Function and the Attainable Set are stored in the on-board computer, it may well happen that the sampled state variable value does not show up in the look-up table. Thus, in order to overcome this key issue due to the fact that the point $\bar{x} \in I\!R^n$ to which the system is steered at a given time is very likely not to be listed in the stored Value Function look-up table, we need to a result yielding estimates of the values of the Value Function at those points.

Proposition 5.5.2 *Assume that the value of V at $\bar{x}$ is not known, and that there is a grid of points G_δ in $I\!R^n$ such that the maximum distance between neighboring points in G_δ is less than $\delta > 0$.*

Then, there is a simplex $S_{\bar{x}} = \{x_i : i = 1, \ldots n+1\} \subset G_\delta$ which are the closest to $\bar{x}$ such that the estimate $\tilde{V}$ of V at $\bar{x}$ is given by

$$\tilde{V}(\bar{x}) = \frac{\sum_{i=1}^{n+1} V_i \|\bar{x} - x_i\|^{-1}}{\sum_{i=1}^{n+1} \|\bar{x} - x_i\|^{-1}}$$

where, for $i = 1, \ldots, n+1$, $V_i = V(x_i) + \nabla V(x_i) \cdot \bar{v}_i$, with $\bar{v}_i = \bar{x} - x_i$ and the $n \times (n+1)$ unknowns of the vectors $\nabla V(x_i)$, $i = 1, \ldots, n+1$ are given as a solution of the set of $n+1$ set of equations

$$\nabla V(x_i) \cdot (\bar{v}_i - \bar{v}_k) = \frac{V(x_k) - V(x_i)}{\|x_i - x_k\|}.$$

Moreover, we have that, for some $c > 0$,

$$\|V(\bar{x}) - \tilde{V}(\bar{x})\| \leq \max_{x_i, x_j \in S_{\bar{x}}} \{|V(x_i) - V(x_j)|\} + c\delta.$$

Proof First, let us observe that in the absence of state constraints, the Value Function is continuous, and, therefore, differentiable in a subset of full measure. Then, for each point $\bar{x}$ in any given subset of $I\!R^n$, it is possible to select a subset $n+1$ points that constitute a simplex and at which the value of V is known, and, at the same time, are the ones closer to $\bar{x}$ than any other one appearing in the look-up table. So the first part of the proof consists in defining an algorithm that, for each un-tabled point $\bar{x}$, produces a set of points satisfying the above requirements. The remaining part of the proof consists in formulating the required intrapolation procedures. Since these are standard steps they will be discussed in a synthetic way.

Let us describe a procedure to find a simplex in $I\!R^n$, that is, a set of $n+1$ points of independent positions (in other words, the n vectors defined by considering one of the points as origin form a linear independent set) whose convex hull contains a given point $\bar{x}$.

Let us be given $\bar{x} \in I\!R^n$ and a $G_\delta \subset I\!R^n$ so that $\bar{x} \notin G_\delta$ where G_δ mentioned above is a countable set of discrete points, $\{x_1, x_2, \ldots : x_i \in I\!R^n\}$ such that: (i) $\forall i$, there is no j with $x_j \in G_\delta$ and $x_j \in x_i + \delta B_1(0)$; and (ii) $\exists \bar{\delta} > \delta$ with $\bar{\delta} - \delta$ small, such that $\left[x_i + \bar{\delta} B_i(0) \setminus \{x_i\}\right] \cap G_\delta$ is a nonempty set.

The procedure to generate the aforementioned simplex S is as follows

1. Initialization: Let $\bar{\delta} = \delta$ and pick $\varepsilon > 0$ small.
2. Let $S = [\bar{x} + \bar{\delta} B_1(0)] \cap G_\delta$.
3. Check whether the elements of S are in an independent position.
4. If not, discard the elements of S for which the distance to $\bar{x}$ is the greatest until all the elements of S are in an independent position.
5. Check whether $\sharp S = n+1$. If yes, stop. otherwise go to 6.
6. Let $\bar{\delta} = \bar{\delta} + \varepsilon$ and goto 2.

Obviously, at some point $\sharp S = n+1$ and all its points will satisfy the stated requirements.

The proof proceeds by taking as G_δ, the set of points at which the value of V is known. Now, in order to estimate the value of V at $\bar{x}$, we simply have to construct a system of $n(n+1)$ equations with $n(n+1)$ unknowns which are quantities that play the role of each partial derivative of V at each one of the points of S. This is a straightforward procedure which is obvious from the set of equations described in the statement of the result.

5.6 Illustration of the Attainable Set and Value Function Computation

5.6.1 Example of the Unicycle

The unicycle model is a very simple and popular example which presents interesting challenges, chiefly among which, it is a non-holonomic system and, moreover, its velocity set fails to be convex.

By applying the ε-Dense Discrete Set approximation we obtain the approximation to the Attainable Set by the cloud of points represented in figure 5.1.

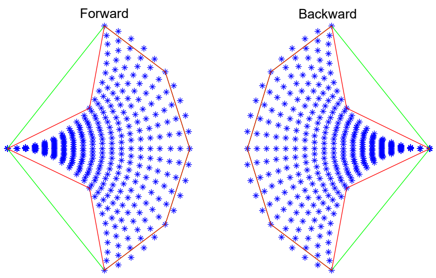


Figure 5.1: Unicycle forward and backward attainable sets

We remark, that, given a cloud of points, it is very straightforward to compute inner, outer or partially inner and outer, possibly nonconvex, polygonal approximation in case of interest. Such a situation may arise, for example, if we need perform set theoretic operations involving sets or regions defined by affine constraints of the type $Ax + b \leq c$. Examples of nonconvex polygons are depicted in figure 5.2.

We take the opportunity for a simple detour from the mainstream flow of this section. The reason for this is simply to make the point that, whenever there are simple methods - possibly defined by a recursive procedure - that are sufficiently efficient to be executed on line to have a good estimate of the Attainable Set, our AS-MPC scheme can easily incorporate it. Below follows an important example for which the Maximum Principle enables such an algorithm.

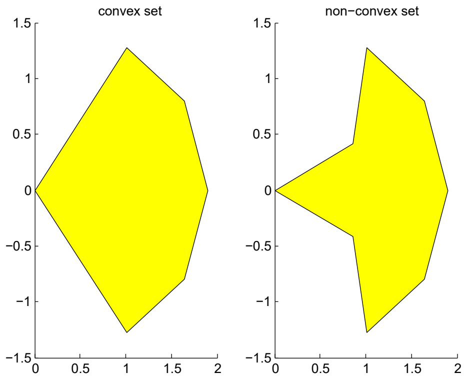


Figure 5.2: Unicycle convex and non convex forward attainable set approximations

Given the special nature and the relevance of the unicycle model, we present an algorithmic approach to exactly compute the Value Function based on the Maximum Principle. Since the approach involves the formulation of a Linear Quadratic Optimal Control Problem, the necessary conditions of optimality are also sufficient and, then the Value Function takes on the optimum values of the cost functional obtained via the Maximum Principle. As it is well known, the Attainable Sets are given as level sets of the Value Function.

Lets consider a coordination transformation on the state space and find the optimal control strategy to the unicycle system by solving the following general (OCP) in the time interval $[t_1, t_2]$.

$$\begin{array}{ll} \text{Minimize} & \dfrac{1}{2} z^T(t_2) Q z(t_2) + \dfrac{1}{2}\displaystyle\int_{t_1}^{t_2} [u(t), w(t)] R [u(t), w(t)]^T dt \\ \text{subject to} & \begin{cases} \dot{z}(t) = A(u) z(t) + B \begin{bmatrix} u \\ w \end{bmatrix}, \quad z(t_1) = z_1 \\ |u| \leq 1, \alpha \leq w \leq 1 \end{cases} \end{array} \tag{5.4}$$

where $R > 0$, $Q \geq 0$, $R = R^T$, $Q = Q^T$.

Following the considerations above, the Value Function at time t and state z is given by:

$$V(t, z) = \frac{1}{2} z^{*T}(t_2) Q z^*(t_2) + \frac{1}{2} \int_t^{t_2} [u^*(t), w^*(t)] R [u^*(t), w^*(t)]^T dt$$

where $(u^*(t), w^*(t), z^*(t))$ is the optimal control process of the above problem.

The considered model is the unicycle

$$\begin{cases} \dot{x}_0 = w\sin(\theta) \\ \dot{y}_0 = w\cos(\theta) \\ \dot{\theta} = u \end{cases} \tag{5.5}$$

where u and w represents the controls, respectively, the turning rate and forward speed, and (x_0, y_0) and θ, are the vehicle's position and orientation. Since this system is nonlinear, a well-known change of variable can be performed to obtain the following unicycle model in new coordinates (Isaac's transformation) (136) which, now, is linear.

$$\dot{z} = \begin{bmatrix} 0 & -u & 0 \\ u & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} \tag{5.6}$$

where the state vector z is $[x, y, \theta]^T$.

Now, we may compute the associated state transition matrix which is given by

$$\begin{aligned} \phi(t_1, t_2) &= e^{\int_{t_1}^{t_2} A(u(\tau))d\tau} \\ &= \begin{bmatrix} \cos(\int_{t_1}^{t_2} u(\tau)d\tau) & -\sin(\int_{t_1}^{t_2} u(\tau)d\tau) & 0 \\ \sin(\int_{t_1}^{t_2} u(\tau)d\tau) & \cos(\int_{t_1}^{t_2} u(\tau)d\tau) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta_2 - \theta_1) & -\sin(\theta_2 - \theta_1) & 0 \\ \sin(\theta_2 - \theta_1) & \cos(\theta_2 - \theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

where $\theta_1 = \theta(t_1)$ and $\theta_2 = \theta(t_2)$.

Thus, the trajectory to system (5.6) in the new coordinates is given by

$$\begin{cases} x(t) = \cos(\theta(t) - \theta_1)x(t_1) - \sin(\theta(t) - \theta_1)y(t_1) + \int_{t_1}^{t} \sin(\theta(t) - \theta(\tau))w(\tau)d\tau \\ y(t) = \sin(\theta(t) - \theta_1)x(t_1) + \cos(\theta(t) - \theta_1)y(t_1) + \int_{t_1}^{t} \cos(\theta(t) - \theta(\tau))w(\tau)d\tau \\ \theta(t) = \theta_1 + \int_{t_1}^{t} u(\tau)d\tau \end{cases} \tag{5.7}$$

Now, we are in position to apply the Maximum Principle of Pontryagin. For that purpose, we start by computing the adjoint system. We obtain the adjoint variable in the following close form:

$$\begin{cases} p_x(t) = p_x(t_2)\cos(\theta_2 - \theta(t)) + p_y(t_2)\sin(\theta_2 - \theta(t)) \\ p_y(t) = -p_x(t_2)\sin(\theta_2 - \theta(t)) + p_y(t_2)\cos(\theta_2 - \theta(t)) \\ p_\theta(t) = p_\theta(t_2), \end{cases} \tag{5.8}$$

which satisfies the boundary conditions:

$$\begin{bmatrix} p_x(t_2) \\ p_y(t_2) \\ p_\theta(t_2) \end{bmatrix} = -Q \begin{bmatrix} x(t_2) \\ y(t_2) \\ \theta(t_2) \end{bmatrix}. \tag{5.9}$$

Now, from the maximum condition off he Maximum Principle, we assert the existence of a vector $\zeta = col(\zeta_u, \zeta_w) \in N_\Omega(u^*, w^*)$, where $\Omega = [-1, 1] \times [\alpha, 1]$ is the control constraint set, that satisfies the relations in the table below.

$$\begin{pmatrix} u \\ w \end{pmatrix} = R^{-1} \left(\zeta(t) + \begin{bmatrix} p_y(t)x(t) - p_x(t)y(t) + p_\theta(t) \\ p_y(t) \end{bmatrix} \right) \tag{5.10}$$

Thus, to compute ζ and the optimal (u, w) we have to check all the primal and dual conditions simultaneously. In particular, (5.10) holds, i.e., if ζ belongs to the normal cone N represented in figure 5.3. This can be done by sequentially checking the conditions that include the ones for $\zeta = \begin{pmatrix} \zeta_u \\ \zeta_w \end{pmatrix}$ listed in Table 5.1 for a given $[u, w]^T$.

Place	**Verification**	**Place**	**Verification**
P_1	$\zeta_u \geq 0$, $\zeta_w \geq 0$	$\overline{P_1 P_2}$	$\zeta_u = 0$, $\zeta_w \geq 0$
P_2	$\zeta_u \leq 0$, $\zeta_w \geq 0$	$\overline{P_2 P_3}$	$\zeta_u \leq 0$, $\zeta_w = 0$
P_3	$\zeta_u \leq 0$, $\zeta_w \leq 0$	$\overline{P_3 P_4}$	$\zeta_u = 0$, $\zeta_w \leq 0$
P_4	$\zeta_u \geq 0$, $\zeta_w \leq 0$	$\overline{P_4 P_1}$	$\zeta_u \geq 0$, $\zeta_w = 0$
Int U	$\zeta_u = 0$, $\zeta_w = 0$		

Table 5.1: Normal cone verifications

Now, we are ready to describe the Algorithm that leads to the computation of the optimal solution to the linear quadratic (OCP) when the trajectory starts at any given initial point. Remark that this algorithm converges in a finite number of steps. Once we have convergence is achieved, the desired value of the Value Function is obtained.

1. <u>Initialization</u>

   - $k = 0$, reset iteration counter

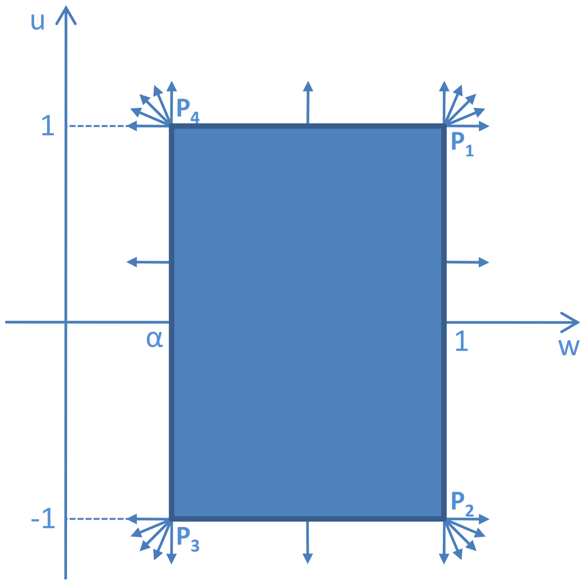


Figure 5.3: Normal cone

- $(u_0, w_0) = (\bar{u}_0, \bar{w}_0) \in U$, define initial control functions (vectors)
- Compute (z_0, p_0) (via (5.7), (5.8) and (5.9))

2. Verification of the optimalilty conditions

   - If (5.10) holds, stop. An optimal control has been found! Otherwise, continue to 3

3. Control update. In order to compute (u_{k+1}, w_{k+1}), find κ such that

$$(u_{k+1}, w_{k+1}) = \text{Proj}_U \left[(u_k, w_k) + \kappa \nabla_{(u,w)} H\right] \tag{5.11}$$

satisfies $J(u_{k+1}, w_{k+1}) < J(u_k, w_k)$, where

$$\nabla_{(u,w)} H = \begin{bmatrix} p_y x - p_x y + p_\theta \\ p_y \end{bmatrix} - R \begin{bmatrix} u \\ w \end{bmatrix}.$$

For this, the following procedure was defined

a) Fix κ sufficiently small

b) Compute $(\bar{u}_{k+1}, \bar{w}_{k+1})$ with (5.11)

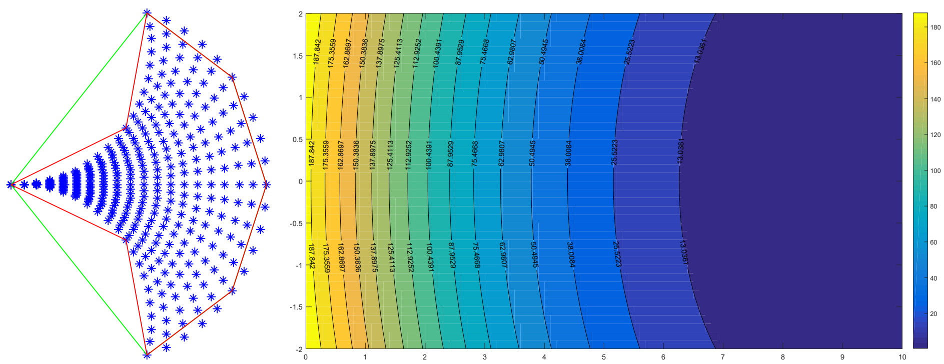
c) Compute $(\bar{z}_{k+1}, \bar{p}_{k+1})$ with (5.7), (5.8) and (5.9)

d) If $J(\bar{u}_{k+1}, \bar{w}_{k+1})$ decreases, increase κ 10% and go to a)

e) Otherwise let (u_{k+1}, w_{k+1}) =$(\bar{u}_{k+1}, \bar{w}_{k+1})$

4. Go to step 2

In Figure 5.4, we represent the Forward Attainable Set and the Value Function for the considered Unicycle system.



(a) Forward attainable set

(b) Value function level sets

Figure 5.4: Unicycle attainable set and value function level sets

Then, the optimal control lookup table for this unicycle is depicted as follows in figure 5.5

The simplicity of the optimization problem is apparent due to the complexity of the computation of the Attainable Set. However, the invariance of the dynamics allows the off-line pre-computation of an approximation of $\mathcal{A}_f(t_0 + T; t_0, x_0)$. In the figure 5.4, it is shown: (i) the forward Attainable Set for the unicycle, and (ii) the Value Function in the absence of obstacles. The Value Function was pre-computed by solving several off-line optimal control problems with different initial conditions spread across a state space partition. Each problem took approximately 3 seconds to compute on ACADO solver on a i7-7500CPU @ 2.70GHz computer and gave rise to a set of trajectories starting from the partition and converging to the final target.

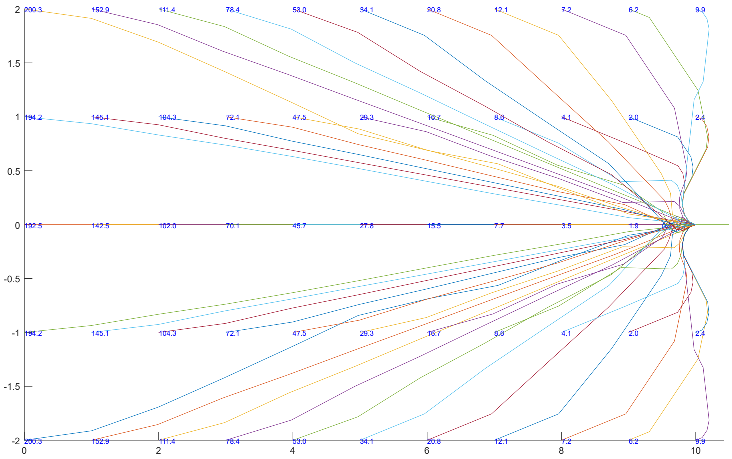


Figure 5.5: Unicycle optimal control lookup table

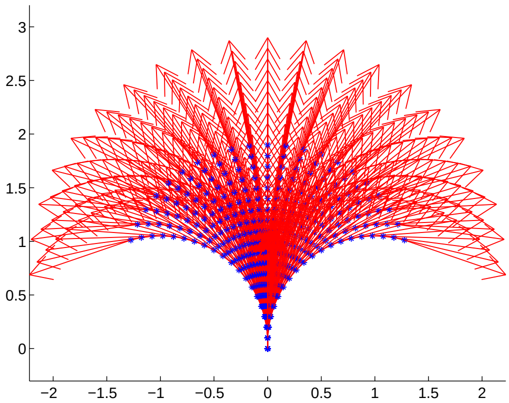


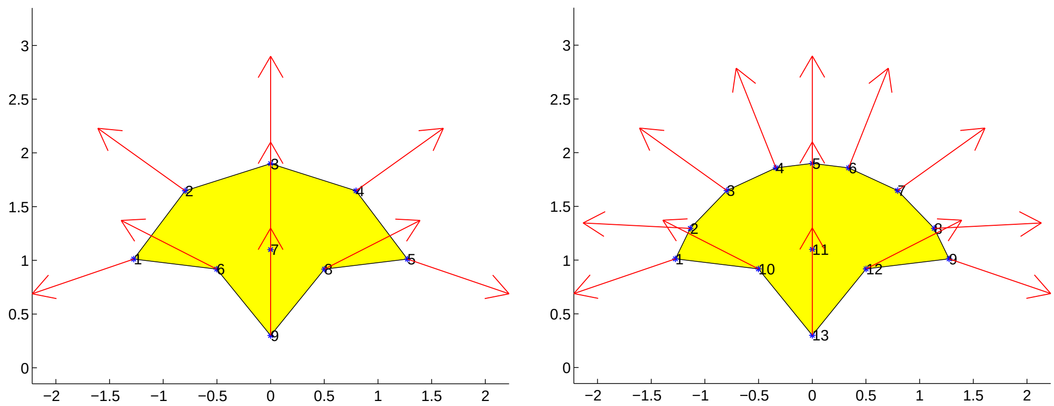
Figure 5.6: Unicycle forward attainable set

5.6.2 Application of the AS-MPC to a specific problem

As a simple aplicaion, assume that we want to drive the unicycle from the initial position $x_0 = [0, 0, 0]$ to a final position $x_f = [10, 6, \pi]$. Let us assume also that the MPC sampling time step is of $2s$ and the vehicle's maximum speed is $1m/s$. The corresponding Attainable Set is depicted in figure 5.6 computed with a time step of $0.1s$.

In order to test the sensitivity, two different discretizations were considered. One with 9 points (figure 5.7(a)) and another with 13 points depicted in (figure 5.7(b)).

Table 5.2 shows the Value Function computed over the time. For instance, at x_0,



(a) Attainable set approximation (9 points) (b) Attainable set approximation (13 points)

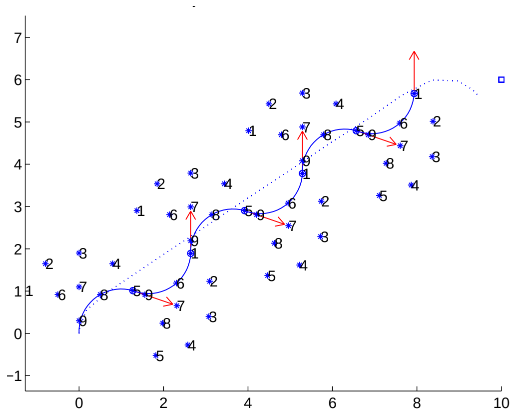
Figure 5.7: Unicycle attainable sets approximations

the cost to go from each of the Attainable Set points $P_i, i = 1..9$ to the target point x_f is listed in the first column. The minimum cost is P_5 and is represented in Green. This point is also chosen to be the initial state for the next MPC computation. In a real scenario eventually with perturbations, the next initial state to be used in the AS-MPC computation would be a real measurement from the vehicle sensors thus providing feedback. The algorithm continues until the target is reached. This execution results can be observed in the figure 5.8.

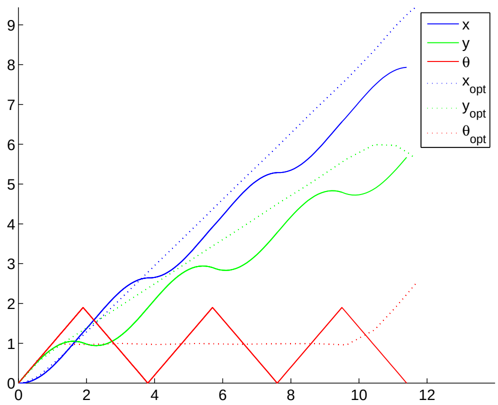
	x_0	x_1	x_2	x_3	x_4	x_5
P_1	856.95	11.17	20.51	5.40	6.59	1.91
P_2	531.54	11.73	11.17	5.81	5.69	2.81
P_3	249.57	25.44	9.39	6.97	4.54	3.79
P_4	119.75	99.45	8.16	8.39	3.59	4.50
P_5	103.84	211.38	7.88	9.51	3.28	4.90
P_6	537.42	23.57	11.11	6.55	5.47	2.71
P_7	343.92	46.93	9.94	7.24	4.72	3.32
P_8	242.49	100.58	9.21	8.16	4.16	3.88
P_9	473.81	82.58	10.68	7.71	5.10	3.28

Table 5.2: Value function over the time

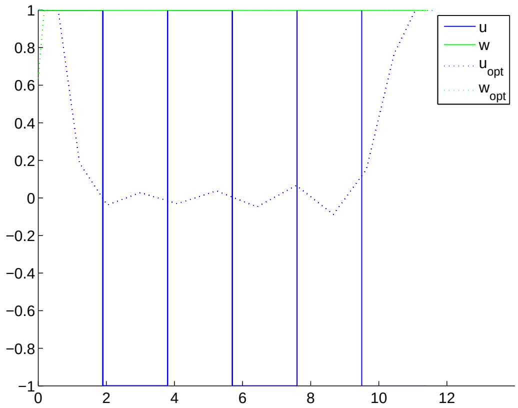
As expected, if more points are used in the approximation, the more the AS-MPC trajectories get closer to the optimal trajectory as we can observe in figure 5.9. The



(a) Trajectories in the XY plane. At each MPC time step, attainable set points are represented in stars, blue circles represent the minimum value function and the blue square represents the target



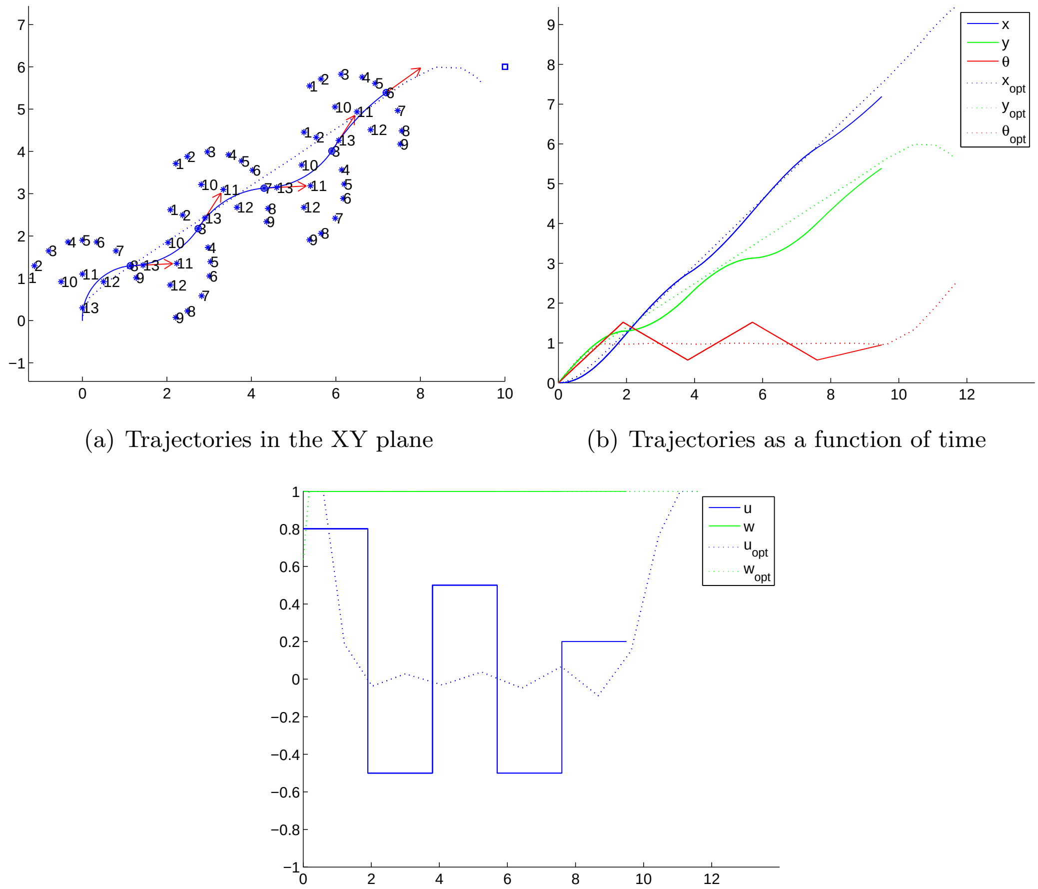
(b) Trajectories as a function of time



(c) Controls as a function of time

Figure 5.8: Attainable set MPC trajectory (solid line) in comparison with the optimal trajectory (dashed line) for a 9 point attainable set discretization

same observation applies to the control functions.



(a) Trajectories in the XY plane

(b) Trajectories as a function of time

(c) Controls as a function of time

Figure 5.9: Attainable set MPC trajectory (solid line) in comparison with the optimal trajectory (dashed line) for a 13 point attainable Set discretization

5.7 Robust Attainable Set MPC scheme

In this section we introduce a variant of the Attainable Set based scheme that ensure robustness to thee presence of short term, low-level and persistent perturbations that affects the dynamics. In this case, The AS-MPC may prove to be too rigid in the sense that, since the system is in open-loop mode during the control horizon, it may well happen that the point in the state space computed previously might not be reachable.

Moreover, if such a behavior happens in a sufficiently large number of control periods, then the AS-MPC may become ineffective. Of course, if a more accurate characterization of uncertainties and perturbations is available, then, in order to ensure the feasibility of the control, a new estimate of the local Attainable Set for which a new approximation will have to be computed is required. On the top of all this, the scheme introduced in the previous section suffers from another major weakness: in networked control of multiple AUVs, if a communication fails, then are no recently computed controls are available, and the only possibility is to proceed with fully simulated data during the whole control horizon.

In order to mitigate these issues a robust version of the AS-MPC, the Robust Attainable Set MPC (RAS-MPC) scheme was introduced in (7) where the optimization in each step is relaxed and the loop is closed within the control horizon Δ with feasible controls. This is to prevent difficulties due to either persistent low level drifting perturbations during the control horizon or missing

In order to overcome these difficulties, we propose a multi-step Attainable Set based scheme enabling a trade-off between complexity, robustness and sub-optimality which can be adjusted to the available on-board resources. This scheme consists in (i) considering a certain sub-optimality for each optimization step, and in (ii) organizing this step in a number of intermediate steps for which only feasibility is required.

Let us assume that γ is a bound on the perturbations exerted on the dynamic control system during the predefined control horizon. Then, a modification of the AS-MPC, designated by RAS-MPC, enabling feedback during the current control horizon, $t_0 + [0, \Delta]$, to mitigate the effect of perturbations during this period is as follows:

1. Initialization: $t_0 = t$, $x(t_0)$.

2. Estimate γ, and let $\Delta_\gamma = \frac{\Delta}{N_\gamma}$ where N_γ is the required number of intermediate samples for feedback.

3. Update $\mathcal{A}_f(t_0 + \Delta; t_0, x(t_0))$ and V at $t_0 + \Delta$.

4. Let z^* be solution to $(P_{\Delta,\gamma})$ which is (P_Δ) modified with $z + \gamma B_1(0) \subset \mathcal{A}_f(t_0 + \Delta; t_0, x(t_0))$

5. Let $t_i^\gamma = t_0 + i\Delta_\gamma$, $I_i^\gamma = [t_{i-1}^\gamma, t_i^\gamma]$ & $\bar{x}_i$ the state sample at t_i^γ. For i=1 to N_γ:

a. Compute z_i s.t. $z_i + \frac{\gamma}{N_\gamma} B_1(0) \subset \mathcal{A}_f(t_i^\gamma; t_{1-1}^\gamma, \bar{x}_{i-1}) \cap \mathcal{A}_b(t_i^\gamma; t_0{+}\Delta, z^*)$.

b. Compute & apply u_i driving the state from $\bar{x}_{i-1}$ to z_i on I_i^γ.

6. Let $x(t_0{+}\Delta) = \bar{x}_{N_\gamma}$, sample of x at $t_0{+}\Delta$, $t_0 = t_0{+}\Delta$, and goto 2.

Again we observe that the forward and backward Attainable Sets required here can be pre-computed, stored on-board and recruited whenever necessary. The intermediate steps seek only to ensure the feasibility of the generated controls to compensate for the perturbations, and, thus, to ensure robustness without having to perform heavy optimization computations. Notice that there is a price to pay for this. The state point z^* computed in step 4. is no longer the optimal point but rather a sub-optimal one in order to provide the flexibility required to accommodate the compensation for the effect of perturbations.

5.8 Conclusions

In this chapter, the main contributions of the thesis were presented. Albeit it has been tested only in a simulation context – a additional results are included in Chapter 6 – this novel AS-MPC scheme has been shown to fulfill the key requirements stated early in this thesis for the control of single AUVs or formation s of multiple AUVs. Key properties such as asymptotic optimality, as well as asymptotic stability have been proved and robustness issues have been discussed. These provide formal guarantees that the conceptual scheme actually performs as the usual MPC scheme but without incurring in the forbidden on-line computational burden that conventional MPC schemes require and makes them unsuitable for the envisaged classes of applications.

Since the ingredients of AS-MPC, the Attainable Set and the Value Function, are complex objects, that are, in fact, are inter-related - Attainable Sets can be regarded as a level sets of the Value Function - the practical implementation of the novel proposed MPC scheme requires the use of some discrete approximation of these objects. With this goal in mind, and after comparing the main existing classes of approximations described in the literature, we proposed a novel type of approximation, the ε-Dense Discrete Set a pproximation, to the Attainable an derived a useful estimation of the Haudorff distance between the exact and approximated Attainable Sets. Moreover, the quality and extent of the approximation refinement is very flexible as it may be obtained by the “discretization” of the control constraint set as well as the number of segments to considered in the piecewise controls used o generation the points in the approximation to the Attainable Set. Moreover, this has a very relevant advantaged in that it produces the controls to be used on-line.

Finally, several illustrations of the approach are shown together with a simple case for which the exact Attainable Set and Value Function can be computed with the help of the Maximum Principle.

Chapter 6

Integration of MPC Scheme in a Control Architecture

6.1 Introduction

These days, most of the control systems of interest are complex in the sense that they might involve the interaction of various subsystems, and exhibit a significant diversity of modes of operation in order to preserve their purpose in the face of significant internal or external changes.

In particular, AUVs are an interesting case in point. The most relevant applications usually require missions with more than one vehicle, possibly controlled in formations which involve the exchange of navigation and payload data. Moreover, their operational environment might exhibit high variability or require the detection of, possibly unexpected, “events”, such as, highly variable spatial-temporal localization of the phenomena whose observation is of interest, emergence of unmapped obstacles or of other features of interest, detection of mines or of intruders in surveillance missions, among many other possibilities.

This means that each AUV - either in a single or in a multiple vehicle missions - should be able to engage in various - often quite diverse - modes of operation, albeit still subordinated to the overall collective mission purpose. The dynamics of the AUV or AUVs, the motion constraints that they have to satisfy, as well as their performance functionals may have to change dramatically from one moment to another triggered by the occurrence of an event. In other words, the conventional dynamic system context do not suffice and an hybrid systems modelling framework has to be adopted.

In this chapter, we show how the AS-MPC (and, as consequence, RAS-MPC) scheme can be run for controlled hybrid systems in the context of single or multiple AUVs, and, in this later case, with a special focus on formations of vehicles. Very much like the context considered in Chapter 4, the either centralized or decentralized architectures will be considered. This new context brings in important formal issues that, in the case of unforeseeable events - which are the really case of practical interest, as, otherwise, a combinatorial framework would trivially reduce this general problem to the one considered in the previous chapter - can hardly be addressed in a satisfactorily general stochastic hybrid control systems framework, as this would require space and effort well beyond the scope of this thesis. The huge challenge of interrelating continuum-time control strategies with controlled discrete events to compensate the effects of either continuum-time and discrete-event, possibly large, disturbances.

Therefore, we will outline a number of practical-driven concepts, and, based on them, methods, to support the analysis and synthesis of hybrid control strategies illustrated in instances of the application problem at hand.

This chapter is organized as follows: In the next section, we introduce and justify the need of considering control architectures, how it emerges from the need of controlling systems in order to accomplish their goals in the presence of both time-continuum dynamics and discrete-event driven trajectories in the course of the execution of the mission. Then, we will focus on modelling hybrid systems via hybrid automata as well as the key automaton property of controllability (non-blocking and liveness). The design of automata controllers will be considered under the following assumptions: (i) the discrete and of the continuum-time controller components driving the system evolution can be "separated", and (ii) the discrete changes have a dominant impact in the evolution of the system. While the former allows us to use, separately, standard synthesis results in either control systems theory and in automata theory, the later provides a frame to define condition under which the discrete synthesis "dominates" the continuum-time synthesis.

Once the dynamics of the system to be controlled and the environment in which its operation are characterized, we will discuss the use of the AS-MPC (and, obviously, the RAS-MPC; for the sake of simplicity, from now on, we will refer to only by AS-MPC) and how this specific implementation of the general conventional MPC scheme fits the purpose of conciliating hybrid feedback control with the requirements of resources optimization inherent to the class of addressed applications.

This chapter is concluded with a discussion of the current observed pitfalls and the identified future developments.

6.2 How the control architecture operates in the context of the AS-MPC

In this section, we discuss the general requirements of the MPC problem at hand that calls for the need of a control architecture in order to articulate discrete-events (controlled or not) and continuum-time controls required to steer the AUV or multiple - possibly interacting - AUVs whose motion is naturally subject to laws of physics.

Thus, the main general reason to consider a control architecture arises from the fact that, in many instances, there is no single standard control synthesis framework to formulate and solve the overall control problem - in this case, the mission to be executed by an AUV or a formation of AUVs - and, thus, it is necessary, to organize it into simpler control problems. For the application problem considered in this thesis, a sample of mode of operations could be defined as follows:

- Controlling each AUV to carry out pre-planned, or replanned, tasks to ensure mission success.
- Management of the motion of a vehicle or a formation of vehicles including maintenance of each one of the pre-defined formation patterns (e.g.,communications connectivity; AUVs role exchange, etc.).
- Adaptation of AUV(s)tasks - in particular, motion patterns - in order to fulfill the specified mission requirements.
- Management of the switching between formation patterns which might depend on expected or unexpected events.

For the sake of simplicity, we will consider the application of the AS-MPC to simple mission scenario for single AUV and for a reconfigurable formation of AUVs modelled by hybrid automata or by networks of hybrid automata. In the case of multiple AUVs, or even systems in which a single AUV interacts with other devices or operators, the AS-MPC scheme will be implemented in a decentralized way. This means, that each AUV will have onboard, besides (i) acoustic communication system when submerged, (ii) electromagnetic communications when at the surface, and (iii) payload and navigation sensors, and (iv) obstacle detection sensors (e.g, range finders), also a (vi) computer system to determine all the tasks to be done in the light of all the information available and the downloaded mission plan. These tasks, include the computation of the next way point (the point - including the orientation θ besides (x, y, z) - at which the vehicle should be after a certain time, from the current instant), detection and characterization of obstacles, payload data gathering strategies, navigation, obstacle collision avoidance strategies, specification of mission roles, localization-driven activities, etc... Obviously, for the AUV motion to take place in the intended manner, its actuators have to receive the appropriate input signals which are generated by the low level controllers, by taking into account both data generated by the vehicles and external perturbations to the vehicle motion.

Before outlining how the AS-MPC enters into the on-board decision making process, let us describe a simple mission scenario that will serve as illustration at later stages.

At some point, the mission execution is initialized and the vehicle systems are recruited and configured in order to execute the tasks composing the mission according to the pre-defined mission plan. For example, the AUV is following a given path along which payload data is gathered according to some strategy.

In the case of the detection of un-expected (and, thus, unmapped) obstacle, say with data from the range finder sensor, the ongoing mission execution has to be changed. By this, it is meant that, besides the on-going activities (such as, data gathering, navigation low level control, and, possibly, others), a new activity of obstacle collision avoidance has to take place in order to guarantee the success of the mission. This activity encompasses the characterization of the obstacle and the redefinition of the motion mode that mitigates the perturbation to the original path while avoiding the collision with the obstacle.

The recruiting of this activity happens because the obstacle detection event caused a transition to a place of the hybrid automaton (which will have to include places for all possible AUV behaviors) at which all the required functional capabilities are activated.

The AUV proceeds with its motion generated by the AS-MPC while the obstacle is not detected. As depicted in figure 6.1, an obstacle is detected by the motion hybrid automaton whenever it falls within the cone of the range sensor represented by the gray triangle.

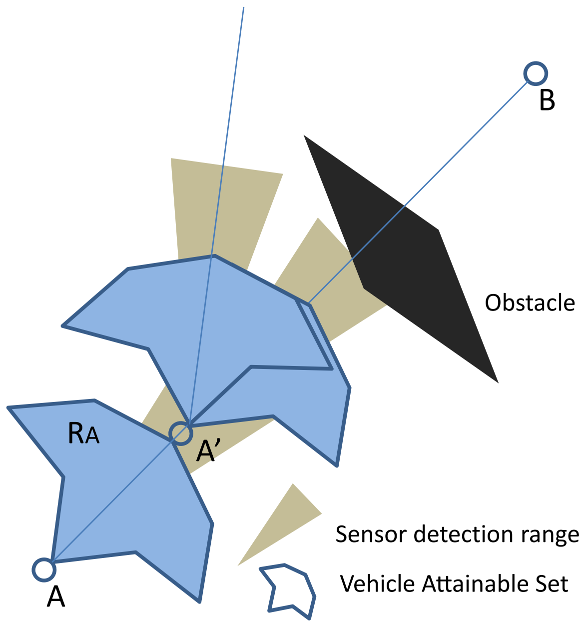


Figure 6.1: Attainable set obstacle detection

Then, once this event happens, the motion hybrid automaton switches to an exploration mode which directs the AUV to proceed in order to find the best way - provided by the AS-MPC in this mode - to circumvent the obstacle while taking into account the original final target. Figure 6.2 illustrates the considerations above.

Thus, at each place, the required dynamics of the AUV (or AUVs) and of the subsystems that are activated there, has to be incorporated in the Attainable Set and, when necessary, in the Value Function of the AS-MPC controller described in the previous chapter. Again, the major computational effort of this incorporation is made off-line, being the "on-line" circumstantial adaptivity of much lower computational complexity. In this way, the proposed MPC controller is always adapted to the multiple phases of the mission with minor additional computational effort required when, even significant changes take place in the course of the mission.

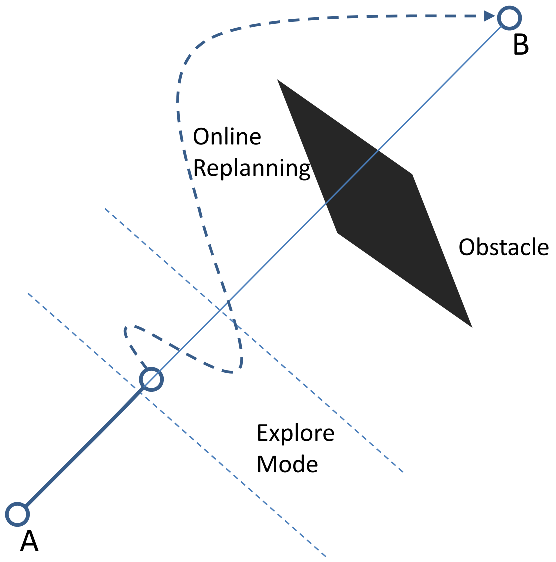


Figure 6.2: Attainable set obstacle avoidance control

It is clear that the above considerations are easily transposed to any other context in which, more or less sudden, changes in the behavior of an AUV or a set of AUVs relatively to the activities of the a priori plan are required due to the occurrence of controlled or uncontrolled discrete events.

From the above, we can regard the proposed AS-MPC as an MPC scheme modified as described in Chapter 5 but now applied to a controlled hybrid system described by an hybrid automaton. This point of view justifies the contents of the next section.

6.3 Brief introduction to controlled hybrid automata

As mentioned before, the term “Hybrid System” designates dynamic systems which are driven by both discrete events and continuum-time dynamics. Hybrid Automata is a quite popular representation of this class of systems due to the fact that, by reflecting a certain decoupling of the discrete and continuous components, of the evolution of the systems, it allows the usage of formal methods of Automata Theory as well as of general control systems with dynamics given by, for example, controlled differential ordinary equations.

The automaton, loosely speaking, involves among other ingredients, a set of places and a set of events triggering the transition between places. That is, one place is active at a time, and, when a certain event occurs, than a new place becomes active while the previous one becomes inactive. The hybrid nature steams from the fact that, in each active place, the state variable of the system evolves in continuum time according to some controlled ordinary differential equation which might change from one place of the automaton to another. It is important to note that the overall state trajectory of the system might or might not exhibit discontinuities at the times when the discrete transitions occur. Moreover, as it will be clear from the formulation below, it might well happen that the continuum-time evolution of the system interferes in the determination of the events triggering the discrete transitions.

More formally, a hybrid automaton H is a collection $H = (Q, X, f, U, Init, D, E, G, R)$, where:

- $Q = \{q1, q2, \cdots\}$ is a set of discrete places.
- $X = I\!R^n$ is a state space, that is the finite dimensional space in which the continuum time state variable evolves in time.
- $f(q, x, u) : Q \times X \times U \to X$ is a vector field.
- $U \subset I\!R^m$ is the, typically compact, set in which the continuum control function $u : I\!R \to I\!R^m$ takes values.
- $Init \subseteq Q \times X$ is a set of initial places and state variable values.
- $Dom(\cdot) : Q \to P(X)$ is a domain of evolution of the state trajectory in a given place.
- $E \subseteq Q \times Q$ is a set of edges, each one associated with an event triggering a transition between different places or for the place.
- $Inv(q)$ is an invariance condition o be satisfied by the continuum-time state variable x at the place q.
- $G(\cdot) : E \to P(X)$ is a guard condition, that forces a transition when x satisfies some given conditions.

- $R(\cdot,\cdot) : E \times X \to P(X)$ is a reset map that assigns values to the state variable x.

Recall that $P(X)$ denotes the power set (set of all subsets) of X. This notation of suggests, for example, that the function $Dom(\cdot)$ assigns a set of continuous states $Dom(q) \subseteq \mathbb{R}^n$ to each discrete state $q \in Q$. $(q, x) \in Q \times X$ is also commonly referred in the hybrid systems literature as the state of H.

There is a vast literature in this field which unfortunately is scattered by a large number of somewhat disconnected communities and, also unfortunately, involving very diverse notation and formalisms. Thus, we opted to cite only one reference, (137), which, very likely, is one of the few that has the virtue of attempting to unify the field with a very significant success. If the need arises, the reader will be rightly directed to specific texts.

In this section we will introduce informally a minimal number of concepts that will be pertinent for the control synthesis in the context of the AS-MPC. To start with we assume that the discrete state transition system is observable in the sense that the initial state of the system (in the hybrid systems sense) can be determined once the observed final state, the continuum-time control and the chain of events are known. This assumption can be relaxed under some circumstances but our goal is to keep the illustration of the AS-MPC in the hybrid systems context as simple as possible.

A controlled hybrid system has no viable control processes within a given time interval $[t_0, T]$ if there exists a feasible $\exists(q, x)$ (i.e., reachable from the initial state) for which there is no finite $N \in \mathbb{N}$ such that $\{q_i, t_i, u_{q_i}(\cdot)\}_0^N$ with $\{t_i\}$ monotonically increasing with $t_N \leq T$ enabling the system to reach the desired final constraint set. If there is a state space set from which there is no hybrid control processes allowing the system trajectory to leave that set then we say that this set is blocking for the given dynamic hybrid control system. It is nonblocking, otherwise. The later property is also known by controllable. We are interested in designing Closed Loop Hybrid Controllers (CLHC) so that the overall system resulting from the composition of the original system with the CLHC includes at least one hybrid control process that drives the system from its initial state to the desired target set. Thus, the control synthesis in the above hybrid systems context consists in designing a hybrid system controller which when composed with the original system, ensures the desired behavioral properties. From this definition, it is clear that there is a strong resemblance with the feedback control for systems whose dynamics are given by ordinary differential equations.

Moreover, the framing of the control design in an optimization context, will provide the basis for the extension of the AS-MPC from conventional dynamic control systems to the hybrid systems context. However, now, besides the usual continuum-time controllers for which there is a vast number of techniques consolidated in a huge number of publications in Control and Systems Theories, we associate an automaton that, on the basis of the current state of the hybrid system, generates a set of new events that triggers transitions in the original hybrid automaton so that the desirable behavior is guaranteed, (137) and references therein. Thus, "closing the loop" here means to compose both hybrid automata. This yields a new automaton which typically exhibits a very high complexity. To avoid this, there is a number of techniques, often supported by software tools, that allows to determine equivalent much simpler equivalent automaton, (137).

6.4 AS-MPC scheme for AUV systems modelled by hybrid automata

As follows from the above, the high variability not only of the environment but also of the context in which the missions being executed may encounter - for example, the spatial and temporal location of the phenomena of interest to be observed, the interference of phenomena whose occurrence was strongly unexpected as well as the multiple types of significant perturbations such currents, internal waves, etc. - the overall control system requires a situational awareness sufficiently expressive to discriminate a set of typified events that will trigger a set of modes of operation that will ensure a significant success in the mission execution.

As mentioned in the introduction of this chapter, the complexity involved in the joint optimization of continuum time measurable controls and controlled discrete-event systems in the context of sense of "optimization adjusted to the highly performance significant uncontrolled discrete events" of the given performance functional is huge and, in fact, there are no practical theoretical results that would support the required formulation of the (OPC) underlying the MPC scheme.

Thus, the alternative that remains is to make use of the already well consolidated Process Systems Engineering methodologies in order to define: (i) all the modes of operation that can occur in the course of the system's life cycle in the context of the set of its purposes; and, for each mode of operation, specify: (ii) the conditions under which the specified requirements are met; (iii) the set of uncontrollable events that might happen and the viable operation modes (if any), possibly depending on specific conditions that have be determined, upon their occurrence; (iv) the continuum-time dynamics and controls as well as the conditions under which they are enabled; and (v) controlled discrete events that can be generated on the basis of the available information, the conditions under which thy are enabled and the modes of operation of the system a result of their occurrence.

This analysis is carried having in mind the overall optimization of the given mission performance criterion. Once, this analysis is carried out we have all the ingredients required to specify an controlled hybrid automaton which is a model representing the system control architecture

As we saw in Section 6.2, the AS-MPC provides the mechanism to select the discrete events o be generated and the continuum-time control at each place of the automaton in order to optimize he overall system performance. In the diagram 6.3 we depict the general (more abstract) automaton presiding the overall behavior of the system.

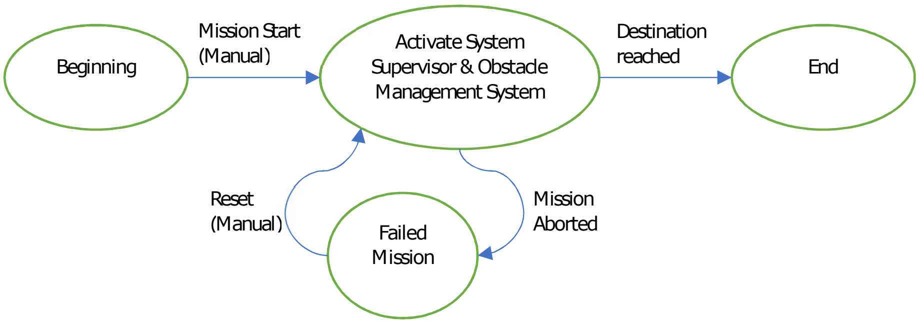


Figure 6.3: Main system automata

This automaton sits at the top of an abstract automaton that encapsulates multiple automata addressing the multiple operation modes. In a compact form, this can be represented as follows in Figure 6.4:

In order to facilitate understanding of the role of the hybrid dynamics in what concerns the obstacle collision avoidance subsystem for a single AUV, whose hybrid dynamics satisfies the rationale depicted in Figure 6.6. From it, it is clear that the AUV proceeds with its motion generated by the AS-MPC scheme while the obstacle is not

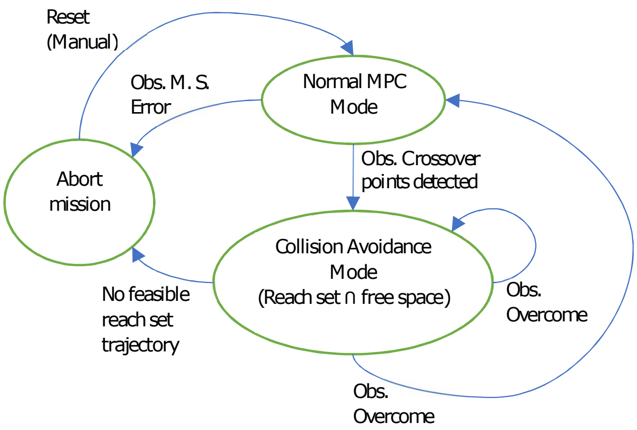


Figure 6.4: System supervisor automata

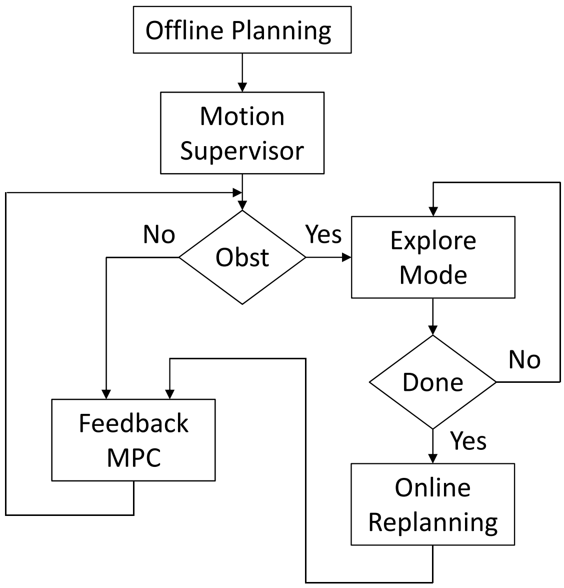


Figure 6.5: Obstacle avoidance control architecture

detected. Once this event happens, the motion supervisor switches on to an exploration mode in order to find the optimal way to circumvent the obstacle by taking into account the original final target which, corresponds to the following control architecture diagram in Figure 6.5.

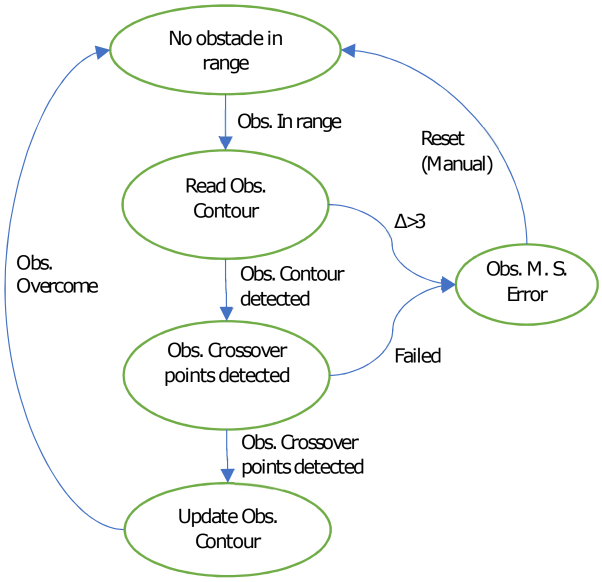


Figure 6.6: Obstacle collision avoidance management

As an illustration, the control mechanism design to generate a controlled discrete event corresponding to the existence of a safe passage between obstacles is based on gathered data gathered in the explore mode and it involves geometric considerations a depicted in Figure 6.7.

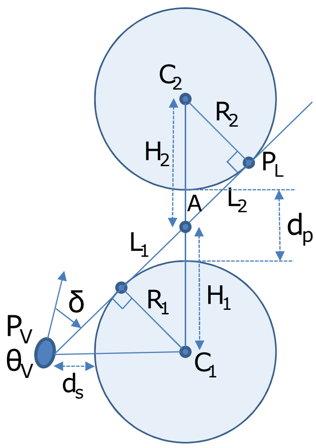


Figure 6.7: Safe passage detection controlled event

The passage is safe if $H_1 + H_2 - R_1 - R_2 - 2d_s > 0$ where d_s is given, R_1, R_2, C_1,

C_2, and P_L are estimated with the range finder, $H_1 = \sqrt{R_1^2 + L_1^2}$, $H_2 = \sqrt{R_2^2 + L_2^2}$, $L_2 = |P_L - A|$, $L_1 = |A - P_V| - |\sqrt{(R_1 + d_s)^2 - R_1^2}|$, P_V is the position of the AUV, and the point A is the intersection of the segments $\overline{C_1, C_2}$ and $\overline{P_V, P_L}$.

Remark that, during the exploration phase, the pre-computed Value Function is used in the search of the path with minimum cost to circumvent the obstacle. Once the exploration activity is successfully terminated, a new path from the current position to the original final target is replanned by using the AS-MPC with an external barrier function added. After the obstacle is circumvented, then, the original pre-computed Value Function can be used in the AS-MPC in order to steer the AUV to the target. This scheme can be easily expanded to an arbitrarily number of imbricated obstacles - i.e., a new obstacle is detected while exploring or trying to overcome a current obstacle.

These ideas can be easily expanded to general AUV operations that properly describe the control architecture. Three modes of operations were considered: normal AS-MPC, Explorer and Collision Avoidance mode. The corresponding automata are depicted in Figures 6.3, 6.6 and 6.4 which present appropriate automata that implement the corresponding logic, choosing the right mode of operation at every stage of the mission.

6.5 Hybrid AS-MPC Simulation Results

Here, we are going to illustrate in simulation of a few paradigmatic general situations how the above AS-MPC scheme operates in the case where the dynamics are given by a controlled hybrid system. Here, we will consider again the issue of controlling the motion of an AUV or a formation of AUVs as the more effective way of convey the features and the challenges underlying the proposed approach.

Lets assume the vehicle's mission is to go from a certain initial to a final location with possible obstacles. Figure 6.8 will be used to illustrate the types of situations the vehicle will experience along the way.

The mission starts with a manual command which activates the System Supervisor and the Obstacle Avoidance system (Figure 6.3), which in turn, sets the normal AS-MPC Mode in the System Supervisor (Figure 6.4) where the controls to be applied to the vehicle are computed by minimizing the Value Function within the vehicles Attainable Set.

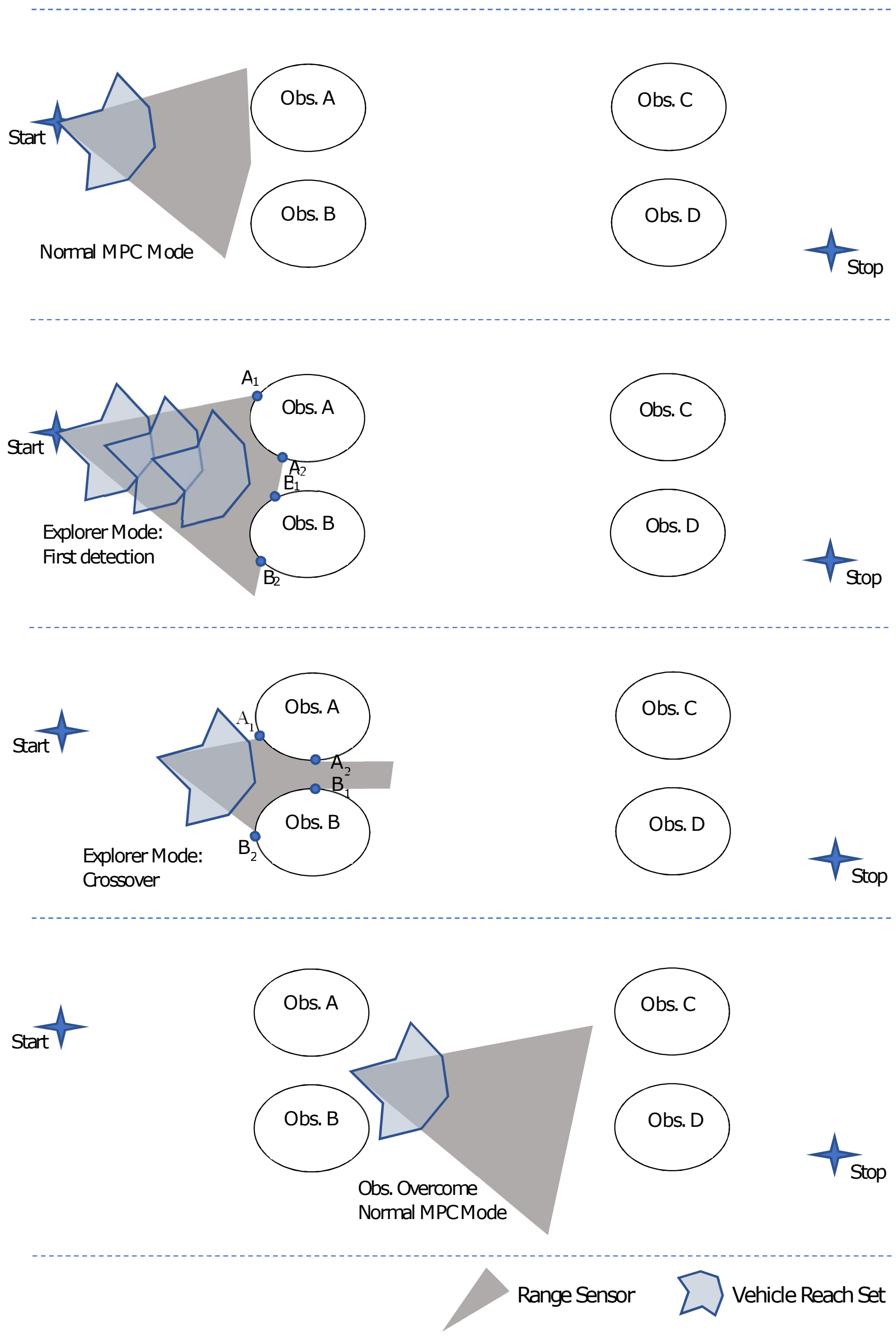


Figure 6.8: Obstacle avoidance illustration

As the vehicle progresses in the pre-computed optimal trajectory and an obstacle is detected, the system goes into explore mode (Figure 6.6) activating the obstacle contour reading task. The vehicle's suite of sensors includes a range sensor with an overture of +/- 45 degrees with respect to x vehicle's fixed frame that performs a complete detection sweep every 5 seconds. For each range sensor detector time step T_{rd}, a contour of the ahead obstacles is built and updated in a SLAM (Simultaneous localization And Mapping) fashion. Each independent contour will define the number of obstacles and its extreme points can be computed and defined as possible crossover points. From Figure 6.8, A_1, A_2, B_1 and B_2 are the points to decide from on where to crossover the obstacles.

At this stage a few assumptions are in order.

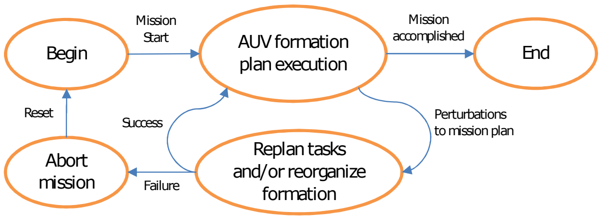
- To allow enough time for the system to detect and avoid obstacles, the range detection must be greater than 5Δ (the MPC control application time).
- An obstacle is considered in range if present for more than $3T_{rd}$ (range sweeps).
- The distance between two consecutive obstacles must be big enough to allow enough clearance for the vehicle or formation of vehicles to successfully crossover.
- For the sake of simplicity, all obstacles must be convex.

If the crossover points of all contours remain fixed for $3T_{rd}$, conclusions with respect to the number of obstacles and possible crossover trajectories can be derived. The decision on where to crossover is as simple as choosing the minimum value of the Value Function. If the contour finding process takes longer than 3Δ or the contour crossover points fails to be detected, an error must be raised and the vehicle's mission must be aborted as no conclusions were found on where to progress.

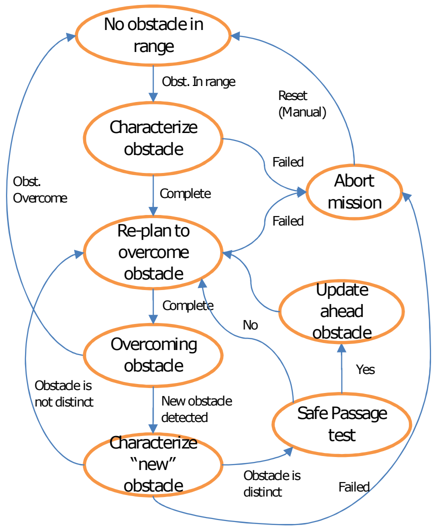
Now that the crossover points are detected, the System Supervisor enters the Collision Avoidance Mode where the current reach set is intersected with the free space leaving a smaller set of trajectories to be used in the crossover. The chosen trajectory must be one that steers the vehicle to the minimum value in the Value Function map. During the crossover, contour updates are required so the intersection of the reach set and the free space is also updated. If no feasible trajectory is available an error must be raised to abort the mission.

Once all obstacles are overcome the system goes back to the tracking mode and the mission will resume until the final destination is reached.

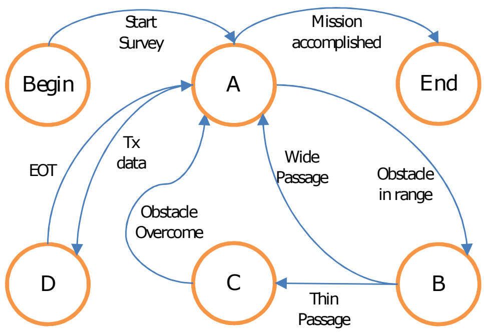
A few more assumptions are also needed:



(a) Main system automaton



(b) Obstacle collision avoidance automaton



(c) Formation Pattern automaton

Figure 6.9: Hybrid system AS-MPC automaton for vehicle formations

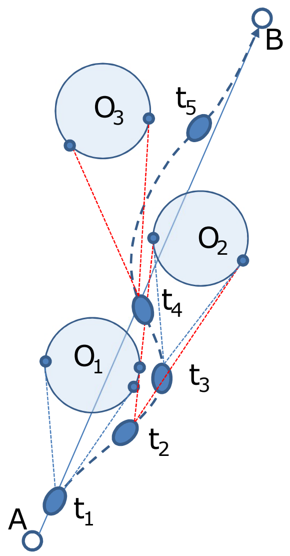
- An obstacle is considered overcome when no obstacle is in range for more than $3T_{rd}$.
- The distance between the just overcome cluster of obstacles and the next must be greater than the range detection to allow the automata to properly evolve.

Figure 6.9(a) shows the overall system automata representing the highest layer of the control architecture. The automaton diagram 6.9(b) shows the various modes and associated transition events. The set of discrete modes associated with these tasks and the events causing the transition between modes are represented by the automaton diagram 6.9(c).

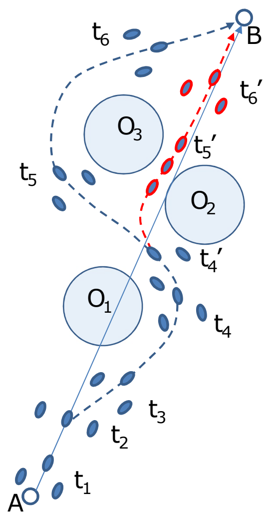
Simulation results obtained with the proposed control structure are shown in figure 6.10. The mission represented in Figure 6.10(a) consists in gathering data by a single AUV while tracking a path defined by the line segment joining points A and B.

At time t_1, obstacle O_1 is detected in the vehicle's path. The Value Function is locally altered around O_1's area by increasing significantly it's cost to keep the vehicle out of it. This forces the vehicle to overcome the obstacle by the right. Since at time t_2 obstacles O_1 and O_2 are in range, and O_1 is the closest obstacle, the Value Function alteration around O_1 is kept while the system decides if there is a safe passage. At time t_3, a safe passage between O_1 and O_2 is detected and the Value Function is now locally altered around O_1 and O_2 to prevent collisions against each obstacle. The path is now chosen by the left of O_2 as it minimizes the Value Function. The same happens at time t_4. A safe passage is detected and the path to B is straightforward. Had the distance between O_2 and O_3 been such that the passage was unsafe, a not-so-optimal solution would have been obtained as the traveled distance by the left of O_3 would be longer than that by the right of O_2.

The mission in Figure 6.10(b) is similar but now with a formation of 3 vehicles in a given triangle formation. In this scenario, the hybrid automaton enabled the adaptivity of the formation pattern or, even, its reconfiguration. In the simulation, we



(a) Single vehicle

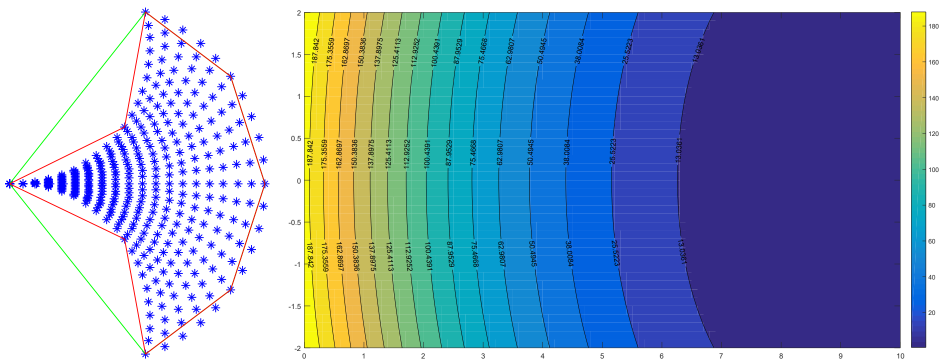


(b) Formation of vehicles

Figure 6.10: Hybrid AS-MPC obstacle avoidance simulations result

considered two different AS-MPC schemes: one with a performance functionals that weights heavily the preservation of the formation pattern, and another weighting more the error in tracking the path defined in the mission. It is interest to see the interplay between performance optimization and the generation of controlled discrete events. In the first the AUVs maintained the formation pattern but had to take a longer path, and thus, with much bigger tracking error, in order to reach the target, while, in the second case, the AUVs switched to a line formation in order to be able to path through a narrow passage that enabled them to reach the target iwith much lower tracking error. This example, clearly illustrates how the AS-MPC is suitable to control hybrid systems by generating hybrid control strategies that contribute to the optimization of the system's performance.

In this section we will present some simulation results in different scenarios including obstacle avoidance. Here we will resume the unicycle example described in Section 5.6.1. In particular, we will use the unicycle Attainable Set and the Value Function presented in Figure 6.11 as ingredients to run the AS-MPC scheme described in 5.3.2. These simulations can be also obtainend in https://paginas.fe.up.pt/~dee04005/attainable-set-mpc/.



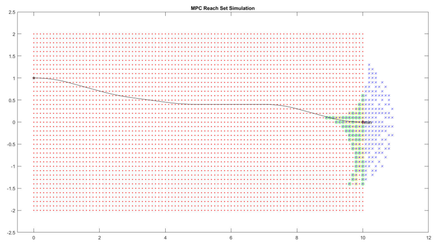
(a) Forward attainable set

(b) Value function level sets

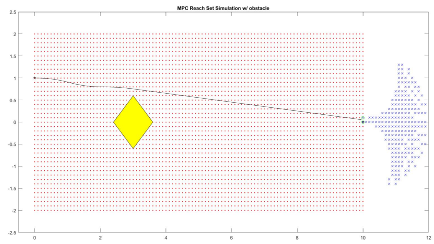
Figure 6.11: Unicycle attainable set and value function level sets

The state space area under consideration is a rectangle of $5m \times 10m$. Depending on the initial condition, the Value Function the system is steered to the target point (10,0,0).

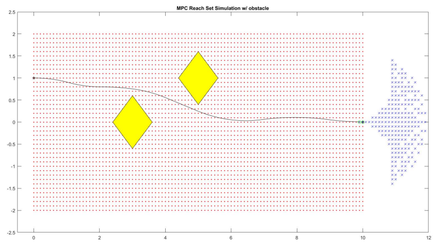
In the first simulation, Figure 6.12(a) we can observe the optimal trajectory sliding down the Value Function. In particular, disturbances such as currents or wind can be compensated by the character of the MPC scheme. Check simulation video.



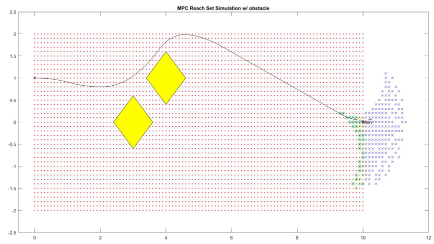
(a) Attainable set MPC with no obstacles



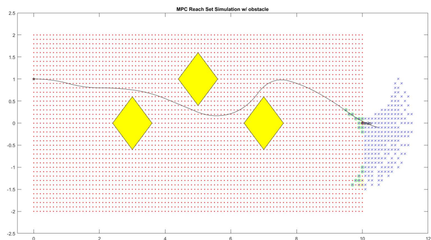
(b) Attainable set MPC (one obstacle)



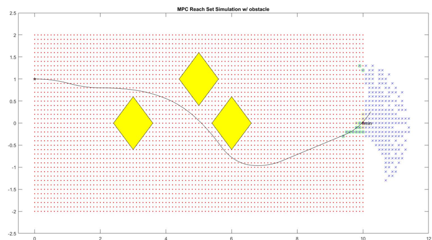
(c) Attainable set MPC (two obstacles)



(d) Attainable set MPC (two close obstacles)



(e) Attainable set MPC (three obstacles)



(f) Attainable set MPC (three close obstacles)

Figure 6.12: Multiple AS-MPC simulations including obstacles

If we now add one obstacle along the trajectory of the system we observe in the Figure 6.12(b) a slight deviation from the optimal trajectory to avoid collisions. The collision avoidance is achieved because there is a portion of the vehicle's Attainable Set that is also part of the obstacle and therefore not available as a feasible trajectory. Check the simulation video.

If we now add another obstacle we can observe another deviation in the trajectory to accommodate for the new obstacle. Refer to Figure 6.12(c) and simulation video.

Now we move the second obstacle too close to the first so that no trajectory is feasible in between. Even though the optimal trajectory would be in between, the system chose to overcome the obstacles around both obstacles in Figure 6.12(d). Check the simulation video.

If we now move the second obstacle way from the first and add a third, we notice the system's trajectory in Figure 6.12(e) tends to be close to the optimal trajectory without obstacles. Check the simulation video.

If the third obstacle is moved too close to the second, the system is forced to overcome the third obstacle via the Value Function minimum which, in this case, is from below. Check Figure 6.12(f) and the simulation video.

So far we have performed simulations with one single vehicle but we can also control formations of vehicles. Here we can see a vehicle formation running across the state space with 3 obstacles. It starts with a triangle formation, then goes to line formation to overcome the tight a set of obstacles (2nd and 3rd) and, once overcome, goes back to the initial triangle formation. Check the simulation simulation video.

6.6 Conclusions and Discussion

In this chapter, some pertinent key issues concerning the application of the MPC scheme o systems whose dynamics are given by hybrid systems exhibiting both controlled and uncontrolled events with significant impact in the performance of the system are raised. The occurrence of performance-significant uncontrollable events raises issues concerning the sense in which the optimization has to be considered. These issues are formally deep and raises many open questions.

The adopted here consists in considered the optimization relative o the specific stochastic realization that effectively took place. This point of view is the easiest one, but, at the same time also a very practical one from the point of view of the applications. One can say that the burden guaranteeing the results of the system operation is transferred from the formal framework to the organization of the operational context and a proper specification of the guaranteed achievable system's performance.

Since MPC - and, in our context the AS-MPC - concerns coupling optimization with feedback, there is the issue of how to efficiently achieve real time optimization in the space of hybrid controls. To the best of our knowledge, there is no general and systematic (in the sense of system's design) way of transferring the huge - due to the highly combinatorial character - computational complexity from the on-line context to the off-line context. The adopted solution consists in using the powerful an well established Systems Process Engineering methodologies to predefine all possible operational modes as well as transition events and conditions under which events can happen and activities can bee executed in order to specify a control architecture modeled in the controlled hybrid automata framework. This will constitute the formal controlled hybrid system underlying our AS-MPC system.

Several simulation examples illustrate the designed AS-MPC scheme to control general hybrid systems with emphasis on the requirements o single AUVs as well as formation off AUVs.

Chapter 7

Conclusions and Open Issues

In this final chapter, we attempt to provide quick overview of the main addressed challenges and a critical analysis of the concepts and results obtained in the course of the underlying research targeting the objectives of the thesis workprogram.

The main general goal of this thesis consists in designing a novel Model Predictive Control (MPC) scheme that does not require the heavy computational effort typically required by the conventional MPC schemes because of the need to solve a certain optimal control at every step of the relatively short control horizon.

This thesis concerns a novel control framework of the Model Predictive Control (MPC) - designated by attainable Set-MPC - type that seeks to conciliate performance optimization and state feedback control under very strict on-line computational constraints. This challenge was strongly motivated by the need of controlling single AUV and formations of AUV systems which due to the complexity of the underwater environment poses tremendous challenges for the design of advanced data gathering systems. The space – required for payload and other devices –, and the energy – required for actuation, sensing, computation, and communication – are at a premium. Moreover, communications, typically merely acoustic, are difficult due to very low data rates, unreliability and high power consumption. This makes the case extremely efficient management of onboard resources and this implies the need of optimization in a context of a state feedback control. The conventional MPC framework manages to conciliate optimization of resources with state feedback control but by paying the prohibitive price associated with the real-time intensive computation inherent to the need of frequently solving optimal control problems.

The key novel idea that pervades throughout the main contributions of the thesis consists in transferring very substantial computational burden associated with the building blocks of the conventional MPC scheme to the off-line stage, by taking advantage of the time invariance of fundamental subsystems. Besides the necessary contextual items, notably the problem statement, challenges analysis, and a directed and commented state-of-the-art review, this thesis includes an in depth assessment of the application of conventional MPC scheme to a simple AUV formation control scenario that encompassed not only software simulation but also hardware-in-the-loop with field data context. The conclusion from this effort is that the on-line computational burden when using very simple control problems (e.g., small formations) is viable for a reasonable performance level but it is not adequate for problems involving more realistic requirements.

This assessment provided the basis to design the novel AS-MPC scheme proposed in this thesis which requires the off-line computation of the Attainable Set and of the system Value Function and their adaptivity in the on-line context with a very small computational effort. Formal results on asymptotic optimality, and asymptotic stability, required to formally ensure the desired properties of the AS-MPC scheme were proved. In many respects, these results are in-line with the corresponding ones for conventional MPC schemes. The weakening the assumptions - which might be possible by the specific structure of the AS-MPC - constitutes an interesting research avenue. It is important to note here, the important contribution that the approximation of the Attainable Sets by a cloud of points with the desired properties. A constructive procedure was designed but there is still room to improve it by increasing its efficiency and enlarging the scope of applications.

This thesis also include a thorough discussion on robustness and computational tractability. This was very much undertaken in line with the work one so far for conventional MPC schemes. However, the geometric character of the novel proposed scheme certainly will enable to examine these issues under new points of view which might shed light on how to unify the apparently disperse results that have been obtained so far for conventional MC schemes

Finally, given the hybrid - that is, discrete event and continuum-time driven - nature of the envisaged class of systems, this thesis also includes an analysis of critical issues arising in this context. Now, even for the AS-MPC scheme, there is a lot of on-line computational effort that can not be transferred to the off-line stage. By resorting to well-established Process Systems Engineering methodologies, an accurate as possible hybrid control system is developed whose a priori decoupling of discrete-event and continuum time components enables to represent the overall system through an hybrid automaton that will provides the controlled dynamics (in a hybrid systems sense) to the AS-MPC (or, obviously, RAS-MPC). There is plenty of room here to examine very efficient computational schemes that allow to adapt the AS-MPC scheme to the event driven dynamics by taking into account the overall optimization whic includes also the continuum-time dynamics. In this respect, it is also important to derive results specifying the conditions under which asymptotic optimality, asymptotic stability and robutness can be guaranteed under aa low on-line computational budget.

The obtained simulation results illustrate how the developed approach works and point out to quite promising future developments.

References

[1] James Blake Rawlings and David Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing Madison, 2015.

[2] Fernando Lobo Pereira, J. Borges de Sousa, R. Gomes, and P. Calado. *A Model Predictive Control Approach to AUVs Motion Coordination*, pages 9–18. Springer International Publishing, Cham, 2015.

[3] R. Gomes, F. Lobo Pereira, and J. Borges de Sousa. **A Model Predictive Control Framework for AUV Formation Control**. In *Proceedings of CONTROLO 2010, Coimbra, Portugal*, September 2010.

[4] Rui Gomes, Pedro Calado, F. Lobo Pereira, and J. Borges de Sousa. **Motion coordination of Autonomous Underwater Vehicles under acoustic communications**. In *Proceedings of IFAC, Milan, Italy*, August 2011.

[5] R. Gomes and F. Pereira. **A Reach Set MPC Scheme for the Cooperative Control of Autonomous Underwater Vehicles**. In *Procs PhysCon 2017, Florence, Italy*, July 2017.

[6] Rui Gomes and Fernando Lobo Pereira. **A Robust Reach Set MPC Scheme for Control of AUVs**. In Anibal Ollero, Alberto Sanfeliu, Luis Montano, Nuno Lau, and Carlos Cardeira, editors, *ROBOT 2017: Third Iberian Robotics Conference*, pages 213–224, Cham, 2018. Springer International Publishing.

[7] R. Gomes and F. Pereira. **A Hybrid Systems Model Predictive Control Framework for AUV Motion Control**. In *Procs ECC 2018, Limassol, Cyprus*, June 12-15 2018.

[8] R. Gomes and F. Pereira. **A Model Predictive Control Scheme for Autonomous Underwater Vehicle Formation Control**. In *Proceedings Controlo 2018, Ponta Delgada, Aores, Portugal*, June 4-6 2018.

[9] Gregory Dudek, Michael R. M. Jenkin, Evangelos Milios, and David Wilkes. **A taxonomy for multi-agent robotics**. *Autonomous Robots*, **3**(4):375–397, Dec 1996.

[10] Y. Uny Cao, Alex S. Fukunaga, and Andrew Kahng. **Cooperative Mobile Robotics: Antecedents and Directions**. *Autonomous Robots*, **4**(1):7–27, Mar 1997.

[11] M. Schneider-Fontan and M. J. Mataric. **Territorial multi-robot task division**. *IEEE Transactions on Robotics and Automation*, **14**(5):815–822, Oct 1998.

[12] Brian P. Gerkey and Maja J Mataric. **A Formal Framework For The Study Of Task Allocation In Multi-Robot Systems**, 2003.

[13] Mehran Mesbahi and Fred Y. Hadaegh. **Formation Flying Control of Multiple Spacecraft via Graphs, Matrix Inequalities, and Switching**. *Journal of Guidance, Control, and Dynamics*, **24**(2):369–377, 2001.

[14] T. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H. Das Nayar, H. Aghazarian, A. J. Ganino, M. Garrett, S. S. Joshi, and P. S. Schenker. **CAMPOUT: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration**. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, **33**(5):550–559, Sept 2003.

[15] Colin R. McInnes. **Autonomous ring formation for a planar constellation of satellites**. *Journal of Guidance, Control, and Dynamics*, **18**(5):1215–1217, 1995.

[16] D. J. Stilwell and B. E. Bishop. **Platoons of underwater vehicles**. *IEEE Control Systems*, **20**(6):45–52, Dec 2000.

[17] P. Encarnacao and A. Pascoal. **Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft**. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, **1**, pages 964–969 vol.1, 2001.

[18] R. Skjetne, S. Moi, and T. I. Fossen. **Nonlinear formation control of marine craft**. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, **2**, pages 1699–1704 vol.2, Dec 2002.

[19] E. Fiorelli, N. E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D. M. Fratantoni. **Multi-AUV control and adaptive sampling in Monterey Bay**. In *2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No.04CH37578)*, pages 134–147, June 2004.

[20] F. Giulietti, L. Pollini, and M. Innocenti. **Autonomous formation flight**. *IEEE Control Systems*, **20**(6):34–44, Dec 2000.

[21] R. W. Beard, J. Lawton, and F. Y. Hadaegh. **A coordination architecture for spacecraft formation control**. *IEEE Transactions on Control Systems Technology*, **9**(6):777–790, Nov 2001.

[22] A. Richards and J. How. **Decentralized model predictive control of cooperating UAVs**. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, **4**, pages 4286–4291 Vol.4, Dec 2004.

[23] T. Balch and R. C. Arkin. **Behavior-based formation control for multirobot teams**. *IEEE Transactions on Robotics and Automation*, **14**(6):926–939, Dec 1998.

[24] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos. **Nonholonomic navigation and control of cooperating mobile manipulators**. *IEEE Transactions on Robotics and Automation*, **19**(1):53–64, Feb 2003.

[25] Akira Okubo. **Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds**. *Advances in Biophysics*, **22**:1 – 94, 1986.

[26] D. Couzin, Jens Krause, Richard James, Graeme D. Ruxton, and Nigel R. Franks. **Collective Memory and Spatial Sorting in Animal Groups**. *Journal of Theoretical Biology*, **218**(1):1 – 11, 2002.

[27] J. R. T. Lawton, R. W. Beard, and B. J. Young. **A decentralized approach to formation maneuvers**. *IEEE Transactions on Robotics and Automation*, **19**(6):933–941, Dec 2003.

[28] Wei Ren and Randal W. Beard. **Virtual Structure Based Spacecraft Formation Control with Formation Feedback**. In *in AIAA Guidance, Navigation, and Control Conference, (Monterey, CA), American Institute of Aeronautics and Astronautics*, pages 2002–4963, 2002.

[29] M. Anthony Lewis and Kar-Han Tan. **High Precision Formation Control of Mobile Robots Using Virtual Structures**. *Autonomous Robots*, **4**(4):387–403, Oct 1997.

[30] M. Egerstedt and Xiaoming Hu. **Formation constrained multi-agent control**. *IEEE Transactions on Robotics and Automation*, **17**(6):947–951, Dec 2001.

[31] J. P. Desai, J. Ostrowski, and V. Kumar. **Controlling formations of multiple mobile robots**. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, **4**, pages 2864–2869 vol.4, May 1998.

[32] J. P. Desai, J. P. Ostrowski, and V. Kumar. **Modeling and control of formations of nonholonomic mobile robots**. *IEEE Transactions on Robotics and Automation*, **17**(6):905–908, Dec 2001.

[33] N. E. Leonard and E. Fiorelli. **Virtual leaders, artificial potentials and coordinated control of groups**. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, **3**, pages 2968–2973 vol.3, 2001.

[34] P. Ogren, E. Fiorelli, and N. E. Leonard. **Cooperative control of mobile sensor networks:Adaptive gradient climbing in a distributed environment**. *IEEE Transactions on Automatic Control*, **49**(8):1292–1302, Aug 2004.

[35] Raymond Kristiansen and Per Johan Nicklasson. **Spacecraft formation flying: A review and new results on state feedback control**. *Acta Astronautica*, **65**(11):1537 – 1552, 2009.

[36] Yueyong Lv, Qinglei Hu, Guangfu Ma, and Jiakang Zhou. **6 DOF synchronized control for spacecraft formation flying with input constraint and parameter uncertainties**. *ISA Transactions*, **50**(4):573 – 580, 2011.

[37] L. Breger, J. How, and A. Richards. **Model predictive control of spacecraft formations with sensing noise**. In *Proceedings of the 2005, American Control Conference, 2005.*, **4**, pages 2385–2390, June 2005.

[38] Haizhao Liang, Jianying Wang, and Zhaowei Sun. **Robust decentralized coordinated attitude control of spacecraft formation**. *Acta Astronautica*, **69**(5):280 – 288, 2011.

[39] T. Balch and R. C. Arkin. **Behavior-based formation control for multirobot teams**. *IEEE Transactions on Robotics and Automation*, **14**(6):926–939, Dec 1998.

[40] Luca Consolini, Fabio Morbidi, Domenico Prattichizzo, and Mario Tosques. **Leader-follower formation control of nonholonomic mobile robots with input constraints**. *Automatica*, **44**(5):1343 – 1349, 2008.

[41] J. Ghommam, M. Saad, and F. Mnif. **Formation path following control of unicycle-type mobile robots**. In *2008 IEEE International Conference on Robotics and Automation*, pages 1966–1972, May 2008.

[42] Yi Liang and Ho-Hoon Lee. **Decentralized formation control and obstacle avoidance for multiple robots with nonholonomic constraints**. In *2006 American Control Conference*, June 2006.

[43] Santiago Garrido, Luis Moreno, and Pedro U. Lima. **Robot formation motion planning using Fast Marching**. *Robotics and Autonomous Systems*, **59**(9):675 – 683, 2011.

[44] E. G. Hernndez-Martnez and E. Aranda-Bricaire. **Multi-agent formation control with collision avoidance based on discontinuous vector fields**. In *2009 35th Annual Conference of IEEE Industrial Electronics*, pages 2283–2288, Nov 2009.

[45] S. Mastellone, D. M. Stipanovic, and M. W. Spong. **Remote Formation Control and Collision Avoidance for Multi-Agent Nonholonomic Systems**. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1062–1067, April 2007.

[46] B. R. Revathy and V. R. Jisha. **Formation control of mobile robots in the presence of disturbance uncertainties**. In *Conf. on Emerging Technological Trends*, pages 1–6, Oct 2016.

[47] D. H. A. Maithripala and S. Jayasuriya. **Radar deception through phantom track generation**. In *Proceedings of the 2005, American Control Conference*, **6**, pages 4102–4106, June 2005.

[48] C. J. Schumacher and Rajeeva Kumar. **Adaptive control of UAVs in close-coupled formation flight**. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, **2**, pages 849–853, 2000.

[49] Haibo Min, Fuchun Sun, and Feng Niu. **Decentralized UAV formation tracking flight control using gyroscopic force**. In *2009 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, pages 91–96, May 2009.

[50] Ben Yun, B. M. Chen, K. Y. Lum, and T. H. Lee. **A leader-follower formation flight control scheme for UAV helicopters**. In *2008 IEEE International Conference on Automation and Logistics*, pages 39–44, Sept 2008.

[51] D. Galzi and Y. Shtessel. **UAV formations control using high order sliding modes**. In *2006 American Control Conference*, pages 6 pp.–, June 2006.

[52] Seungkeun Kim and Youdan Kim. **Three dimensional optimum controller for multiple UAV formation flight using behavior-based decentralized approach**. In *2007 International Conference on Control, Automation and Systems*, pages 1387–1392, Oct 2007.

[53] Z. Chao, L. Ming, Z. Shaolei, and Z. Wenguang. **Collision-free UAV formation flight control based on nonlinear MPC**. In *Conf. on Electronics, Communications and Control*, pages 1951–1956, Sept 2011.

[54] S. H. Mathisen, T. I. Fossen, and T. A. Johansen. **Non-linear model predictive control for guidance of a fixed-wing UAV in precision deep stall landing**. In *Conf.on Unmanned Aircraft Systems*, pages 356–365, June 2015.

[55] S. Quintero, D. Copp, and J. Hespanha. **Robust UAV coordination for target tracking using output-feedback model predictive control with moving horizon estimation**. In *American Control Conf.*, pages 3758–3764, July 2015.

[56] S. Bertrand, J. Marzat, H. Piet-Lahanier, A. Kahn, and Y. Rochefort. **MPC Strategies for Cooperative Guidance of Autonomous Vehicles**. In *AerospaceLab Journal, Issue 8*, December 2014.

[57] R. Andrade, G. Raffo, and J. Normey-Rico. **Model predictive control of a tilt-rotor UAV for load transportation**. In *European Control Conf.*, pages 2165–2170, June 2016.

[58] Y. Kuriki and T. Namerikawa. **Formation control with collision avoidance for a multi-UAV system using decentralized MPC and consensus-based control**. In *European Control Conf.*, pages 3079–3084, July 2015.

[59] Rongxin Cui, Shuzhi Sam Ge, Bernard Voon Ee How, and Yoo Sang Choo. **Leaderfollower formation control of underactuated autonomous underwater vehicles**. *Ocean Engineering*, **37**(17):1491 – 1502, 2010.

[60] D. B. Edwards, T. A. Bean, D. L. Odell, and M. J. Anderson. **A leader-follower algorithm for multiple AUV formations**. In *2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No.04CH37578)*, pages 40–46, June 2004.

[61] Li W. Xu H. Feng X. Kang, X. **Formation control of multiple AUVs for moth-inspired plume tracing**. **2**, pages 419–424, 2010.

[62] Rui Zhao, X. Xiang, C. Yu, and Zheng Jiang. **Coordinated formation control of autonomous underwater vehicles based on leader-follower strategy**. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–5, Sept 2016.

[63] Qiuling Jia and Guangwen Li. **Formation Control and Obstacle Avoidance Algorithm of Multiple Autonomous Underwater Vehicles (AUVs) Based on Potential Function and Behavior Rules**. In *2007 IEEE International Conference on Automation and Logistics*, pages 569 –573, Aug. 2007.

[64] H. Yang and F. Zhang. **Geometric formation control for autonomous underwater vehicles**. In *2010 IEEE International Conference on Robotics and Automation*, pages 4288–4293, May 2010.

[65] E. Yang and D. Gu. **Nonlinear Formation-Keeping and Mooring Control of Multiple Autonomous Underwater Vehicles**. *IEEE/ASME Transactions on Mechatronics*, **12**(2):164–178, April 2007.

[66] R. Cui, D. Xu, and W. Yan. **Formation Control of Autonomous Underwater Vehicles under Fixed Topology**. In *2007 IEEE International Conference on Control and Automation*, pages 2913–2918, May 2007.

[67] Weisheng Yan, Rongxin Cui, and Demin Xu. **Formation control of underactuated autonomous underwater vehicles in horizontal plane**. In *2008 IEEE International Conference on Automation and Logistics*, pages 822–827, Sept 2008.

[68] C. Shen, Y. Shi, and B. Buckham. **Model predictive control for an AUV with dynamic path planning**. In *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 475–480, July 2015.

[69] Chao Shen, Y. Shi, and B. Buckham. **Path-following control of an AUV using multi-objective model predictive control**. In *2016 American Control Conf. (ACC)*, pages 4507–4512, July 2016.

[70] R. Vinter. *Optimal Control*. Birkhauser, 2000.

[71] W. Langson, I. Chryssochoos, S.V. Rakovic, and D.Q. Mayne. **Robust model predictive control using tubes**. *Automatica*, **40**(1):125 – 133, 2004.

[72] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. **Constrained model predictive control: Stability and optimality**. *Automatica*, **36**(6):789 – 814, 2000.

[73] D.Q. Mayne, S.V. Rakovic, R. Findeisen, and F. Allgower. **Robust output feedback model predictive control of constrained linear systems: Time varying case**. *Automatica*, **45**(9):2082 – 2087, 2009.

[74] Andrea Boccia, Lars Grüne, and Karl Worthmann. **Stability and feasibility of state-constrained linear MPC without stabilizing terminal constraints** . In *MTNS 2014*, Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems, pages 453–460, Los Angeles, United States, 2014.

[75] L. Grüne, J. Pannek, and K. Worthmann. **A networked unconstrained nonlinear MPC scheme**. In *2009 European Control Conference (ECC)*, pages 371–376, Aug 2009.

[76] H. Chen and F. Allgöwer. *Nonlinear Model Predictive Control Schemes with Guaranteed Stability*, pages 465–494. Springer Netherlands, Dordrecht, 1998.

[77] Lars Grüne and Vryan Gil Palma. **Robustness of performance and stability for multistep and updated multistep MPC schemes**.

[78] Alberto Bemporad and Manfred Morari. **Control of systems integrating logic, dynamics, and constraints**. *Automatica*, **35**(3):407 – 427, 1999.

[79] T. Geyer, G. Papafotiou, and M. Morari. **Model Predictive Control in Power Electronics: A Hybrid Systems Approach**. pages 5606–5611, Dec 2005.

[80] F. A. C. C. Fontes and L. Magni. **Min-max model predictive control of nonlinear systems using discontinuous feedbacks**. *IEEE Transactions on Automatic Control*, **48**(10):1750–1755, Oct 2003.

[81] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas. **Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations**. *IEEE Transactions on Control Systems Technology*, **16**(1):19–33, Jan 2008.

[82] H. Michalska and D. Q. Mayne. **Robust receding horizon control of constrained nonlinear systems**. *IEEE Transactions on Automatic Control*, **38**(11):1623–1633, Nov 1993.

[83] Scokaert Pierre O. M. and Rawlings James B. **Feasibility issues in linear model predictive control**. *AIChE Journal*, **45**(8):1649–1659, 2004.

[84] L. CHISCI, A. LOMBARDI, E. MOSCA, and J. A. ROSSITER. **State-space approach to stabilizing stochastic predictive control**. *International Journal of Control*, **65**(4):619–637, 1996.

[85] E. Franco, T. Parisini, and M. M. Polycarpou. **Cooperative control of discrete-time agents with delayed information exchange: A receding-horizon approach**. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, **4**, pages 4274–4279 Vol.4, Dec 2004.

[86] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. M. Raimondo. **Cooperative Constrained Control of Distributed Agents With Nonlinear Dynamics and Delayed Information Exchange: A Stabilizing Receding-Horizon Approach**. *IEEE Transactions on Automatic Control*, **53**(1):324–338, Feb 2008.

[87] Tams Keviczky, Francesco Borrelli, and Gary J. Balas. **Decentralized receding horizon control for large scale dynamically decoupled systems**. *Automatica*, **42**(12):2105 – 2115, 2006.

[88] J. A. Fax and R. M. Murray. **Information flow and cooperative control of vehicle formations**. *IEEE Transactions on Automatic Control*, **49**(9):1465–1476, Sept 2004.

[89] Ulf Jönsson, Chung-Yao Kao, and Hisaya Fujioka. **A Popov criterion for networked systems**. *Systems & Control Letters*, **56**(9):603 – 610, 2007.

[90] P. A. Cook. **Conditions for string stability**. *Systems & Control Letters*, **54**(10):991 – 998, 2005.

[91] M. Allen, J. Ryan, C. Hanson, and J. Parle. **String stability of a Linear Formation Flight Control System**. In *Proceedings of 2002 AIAA Guidance, Navigation, and Control Conference*, number Monterey, CA, 2002.

[92] Xiangheng Liu, A. Goldsmith, S. S. Mahal, and J. K. Hedrick. **Effects of communication delay on string stability in vehicle platoons**. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pages 625–630, 2001.

[93] R. Olfati-Saber and R. M. Murray. **Consensus problems in networks of agents with switching topology and time-delays**. *IEEE Transactions on Automatic Control*, **49**(9):1520–1533, Sept 2004.

[94] E. Semsar-Kazerooni and K. Khorasani. **Optimal consensus algorithms for cooperative team of agents subject to partial information**. *Automatica*, **44**(11):2766 – 2777, 2008.

[95] G. C. Goodwin, H. Haimovich, D. E. Quevedo, and J. S. Welsh. **A moving horizon approach to Networked Control system design**. *IEEE Transactions on Automatic Control*, **49**(9):1427–1445, Sept 2004.

[96] Fernando A. C. C. Fontes, Dalila B. M. M. Fontes, and Amélia C. D. Caldeira. **Model Predictive Control of Vehicle Formations**. In Michael J. Hirsch, Clayton W. Commander, Panos M. Pardalos, and Robert Murphey, editors, *Optimization and Cooperative Control Strategies*, pages 371–384, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[97] L. Grüne, J. Pannek, and K. Worthmann. **A networked unconstrained nonlinear MPC scheme**. pages 371–376, Aug 2009.

[98] Lars Grüne, Frank Allgöwer, Rolf Findeisen, Jörg Fischer, Dominic Gross, Uwe D. Hanebeck, Benjamin Kern, Matthias A. Müller, Jürgen Pannek, Marcus Reble, Olaf Stursberg, Paolo Varutti, and

KARL WORTHMANN. **Distributed and networked model predictive control**. In *Lunze, Jan (Hrsg.): Control theory of digitally networked dynamic systems. Chapter 4.*, pages 111–167. Springer, 2014.

[99] LARS GRÜNE AND JÜRGEN PANNEK. *Nonlinear Model Predictive Control : Theory and Algorithms. 2nd Edition.* Communications and Control Engineering. Springer-Verlag London, 2011.

[100] T. I. FOSSEN. *Guidance and Control of Ocean Vehicles.* John Wiley and Sons Ltd., 1994.

[101] A. J. HEALEY AND D. LIENARD. **Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles**. *IEEE Journal of Oceanic Engineering*, **18**(3):327–339, Jul 1993.

[102] R. M. F. GOMES, J. B. SOUSA, AND F. L. PEREIRA. **Modeling and Control of the IES project ROV**. In *2003 European Control Conference (ECC)*, pages 3424–3429, Sept 2003.

[103] R. M. F. GOMES, A. MARTINS, A. SOUSA, J. B. SOUSA, S. L. FRAGA, AND F. L. PEREIRA. **A new ROV design: issues on low drag and mechanical symmetry**. In *Europe Oceans 2005*, **2**, pages 957–962 Vol. 2, June 2005.

[104] RUI M. F. GOMES. *Modelização e Controlo de Veículos Subaquáticos.* Master's thesis, Porto University, 2002.

[105] E. LEWIS, editor. *Principles of Naval Architecture.* Society of Naval Architects and Marine Engineers, 1989. 2nd revision.

[106] T. J. PRESTERO. *Verification of a six-degree of freedom simulation model for the REMUS AUV.* Master's thesis, Massachusetts Institute of Technology / Woods Hole Oceanographic Institution, Departments of Ocean and Mechanical Engineering, 2001.

[107] MATTHIAS A. MÜLLER AND LARS GRÜNE. **Economic model predictive control without terminal constraints for optimal periodic behavior**. *Automatica*, **70**:128–139, August 2016.

[108] B. Kafash, A. Delavarkhalafi, and S.M. Karbassi. **Application of Chebyshev polynomials to derive efficient algorithms for the solution of optimal control problems**. *Scientia Iranica*, **19**(3):795 – 805, 2012.

[109] B. Kafash, A. Delavarkhalafi1, S. M. Karbassi, and K. Boubaker. **A Numerical Approach for Solving Optimal Control Problems Using the Boubaker Polynomials Expansion Scheme**. *Journal of Interpolation and Approximation in Scientific Computing*, **2014**(18), 2014.

[110] Claire Walton, Chris Phelps, Qi Gong, and Isaac Kaminer. **A Numerical Algorithm for Optimal Control of Systems with Parameter Uncertainty**. *IFAC-PapersOnLine*, **49**(18):468 – 475, 2016. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016.

[111] Z. Rafiei, B. Kafash, and S. M. Karbassi. **A new approach based on using Chebyshev wavelets for solving various optimal control problems**. *Computational and Applied Mathematics*, Feb 2017.

[112] Nikolai N. Krasovskii. *Control of a Dynamical System.* Nauka, Moscow, 1985.

[113] N.N. Krasovskii and A.I. Subbotin. *Game-Theoretical Control Problems.* Springer-Verlag New York, Dordrecht, 1988.

[114] Andrew N. Krasovskii and Nikolai N. Krasovskii. *Control Under Lack of Information.* Birkhuser Basel, Dordrecht, 1995.

[115] Martino Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations.* Birkhauser, 1997.

[116] F.H. Clarke. **A proximal characterization of the reachable set**. *Systems & Control Letters*, **27**(3):195 – 197, 1996.

[117] F. Clarke, Yu. Ledyaev, R. Stern, and P. Wolenski. *Nonsmooth Analysis and Control Theory.* Springer Verlag, New York, 1998.

[118] A. Kurzhanski and P. Varaiya. **Dynamic Optimization for Reachability Problems**. *Journal of Optimization Theory and Applications*, **108**(2):227–251, Feb 2001.

[119] A. Kurzhanski and P. Varaiya. *The Hamilton-Jacobi Type Equations for Nonlinear Target Control and Their Approximation*, pages 77–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[120] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science (Cambridge ... on Applied and Computational Mathematics)*. Cambridge University Press, 2 edition, 1999.

[121] I. Michel, A. Bayen, and C. Tomlin. **Computing Reachable Sets for Continuous Dynamics Games Using Level Sets Methods**. *IEEE Trans. on Automatic Control*, **50**(7):980–1001, 2005.

[122] Ian M. Mitchell. **The Flexible, Extensible and Efficient Toolbox of Level Set Methods**. *Journal of Scientific Computing*, **35**(2-3):300–329, June 2008.

[123] E. A. Cross and I. M. Mitchell. **Level Set Methods for Computing Reachable Sets of Systems with Differential Algebraic Equation Dynamics**. In *2008 American Control Conference*, **31**, pages 2260–2265, June 2008.

[124] R. Bellman. *Dynamic programming.* Princeton University Press, 1957.

[125] T. Graettinger and B. Krogh. **Hyperplane Method for Reachable State Estimation for Linear Time-Invariant Systems**. *Journal of Optimization Theory and Applications*, **69**(3):555–588, Jun 1991.

[126] P. Varaiya. **Reach set computation using optimal control**. In *Proc. KIT Workshop*, 1998.

[127] P. Wolenski. **The Exponential Formula for the Reachable Set of a Lipschitz Differential Inclusion**. *SIAM Journal on Control and Optimization*, **28**(5):1148–1161, 1990.

[128] H. Chen and F. Allgöwer. **A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability****This paper was

not presented at any IFAC meeting. This paper was accepted for publication in revised form by Associate Editor W. Bequette under the direction of Editor Prof. S. Skogestad. *Automatica*, **34**(10):1205 – 1217, 1998.

[129] LARS GRÜNE AND JÜRGEN PANNEK. *Nonlinear Model Predictive Control : Theory and Algorithms*. Communications and Control Engineering. Springer, 2017.

[130] ROLF FINDEISEN AND FRANK ALLGÖWER. **An Introduction to Nonlinear Model Predictive Control**. In *21ST Benelux Meeting on Systems and Control, Veidhoven*, page 119141, 2002.

[131] A.B. KURZHANSKI AND P. VARAIYA. *Ellipsoidal Techniques for Reachability Analysis*. 2000.

[132] A.B. KURZHANSKI AND P. VARAIYA. **On Ellipsoidal Techniques for Reachability Analysis. Part II: Internal Approximations Box-valued Constraints**. *Optimization Methods and Software*, **17**(2):207–237, 2002.

[133] V. BATURIN, E. GONCHAROVA, F. LOBO PEREIRA, AND J. SOUSA. **Measure-controlled dynamic systems: Polyhedral approximation of their reachable set boundary**. *Journal Automation & Remote Control*, **67**(3):350–360, Mar 2006.

[134] P. VARAIYA. *Notes on Optimization*. Van Nostrand Reinhold notes on system sciences. Van Nostrand Reinhold Co., 1972.

[135] ASHISH TIWARI AND GAURAV KHANNA. *Nonlinear Systems: Approximating Reach Sets*, pages 600–614. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[136] VALERII S. PATSKO AND VARVARA L. TUROVA. **From Dubins' car to Reeds and Shepp's mobile robot**. *Computing and Visualization in Science*, **12**(7):345–364, Sep 2009.

[137] S. LAFORTUNE C. G. CASSANDRAS. *Introduction to Discrete Event Systems*. Springer-Verlag, 2008. 2nd edition.

[138] M. Crandall and P.-L. Lyons. **Viscosity Solutions of Hamilton-Jacobi Equations**. *Trans. American Mathematical Society*, **277**(1):487–502, 1983.

[139] A. Hu, J. Dill, A. Drexler, and C. Yang. **Higher-level specification and verification with BDDs**. pages 82–95, 1993.

[140] J. Sethian and A. Vladimirsky. **Ordered Upwind Methods for Static Hamilton–Jacobi Equations: Theory and Algorithms**. *SIAM Journal on Numerical Analysis*, **41**(1):325–363, 2003.

[141] Ian Mitchell, Alexandre M. Bayen, and Claire J. Tomlin. **Validating a Hamilton-Jacobi Approximation to Hybrid System Reachable Sets**. In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, pages 418–432, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[142] Jean-Pierre Aubin. *Viability Theory, Systems and Control: Foundations and Applications*. Birkhauser, Cambridge, MA, USA, 1991.

[143] P. Saint-Pierre. **Approximation of the viability kernel**. *Applied Mathematics and Optimization*, **29**(2):187–209, Mar 1994.

[144] A. J. Healey. **Dynamics of Marine Vehicles**. Lecture notes, Naval Postgraduate School, Monterey, California, 2010.

[145] Lennart Ljung. *System Identification - Theory For the User*. PTR Prentice Hall, 2nd ed edition, 1999.

[146] Joseph J. Keller. *Tracking Control Of Autonomous Underwater Vehicles*. Master's thesis, Naval Postgraduate School, 2002.

[147] P. S. Dias, S. L. Fraga, R. M. F. Gomes, G. M. Goncalves, F. L. Pereira, J. Pinto, and J. B. Sousa. **Neptus - a framework to support multiple vehicle operation**. In *Europe Oceans 2005*, **2**, pages 963–968, June 2005.

[148] P. S. Dias, R. M. F. Gomes, J. Pinto, G. M. Goncalves, J. B. Sousa, and F. L. Pereira lixo. **Mission planning and specification in the Neptus**

framework. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3220–3225, May 2006.

Appendix A

Attainable Sets

The Attainable Set of a dynamical control system – either, discrete, or evolving in continuum time, or hybrid – represents the points of the state space that can be reached by using all the available controls while satisfying the existing state and/or control constraints. If we denote the state transition operator by $\Phi : \mathbf{R} \times \mathbf{R} \times \mathbf{R}^n \times \mathcal{U} \to \mathbf{R}^n$, where $\mathcal{U} = \{u : [t_0, t_f] \to \mathbf{R}^m, u(t) \in \Omega, u(\cdot) \text{ is measurable}\}$ denotes the set of controls available in the specified time interval, we have that the Attainable Set at time t_f from the state x_0 at time t_0 is, in the absence of additional constraints, defined by

$$\mathcal{A}(t_f; t_0, x_0) = \{z \in \mathbf{R}^n : z = \Phi(t_f; t_0, x_0, u), \forall u \in \mathcal{U}\}.$$

This is in fact the notion of Forward Attainable Set from a point and its extension to that from a given set is straightforward. There is also the concept of Backward Attainable Set that specifies the set of points from which a given target point or set in the state space can be reached in a given time interval. To facilitate the exposition, we will focus only on the former.

Attainable Sets are extremely relevant for control and verification. This steams from a number of reasons, of which we would like to single out the following:

a) It enables to prove that a dynamic system reaches a given target while remaining within a specified set. This is important to verify a number of properties, namely, whether the state remains within a given desirable set or enters a forbidden set in which the integrity of the system might be threatened.

b) Since it encompasses systems with control inputs, it can be used to design scheduled or hybrid controllers.

c) For optimal control problems with a cost functional depending only on the state at the final time, the Attainable Set enables to replace the original infinite dimensional optimization problem by a finite dimensional one, i.e., (P) Minimize $\{ g(z) : z \in \mathcal{A}(t_f; t_0, x_0)\}$.

d) It is easily amenable to the incorporation of perturbations and uncertainties which might be expressed in terms of non-controlled inputs. Worst case control strategies can be computed by considering min-max optimization problems.

e) Along the vein of d), differential games with adversarial players can be easily formulated and analyzed by using Attainable Sets.

In the Dynamic Optimization literature, see (118, 119) among others, it has been long established that the Attainable Set can be characterized as a level set, of a solution to a certain Hamilton-Jacobi partial differential equation (HJE). That is, the Attainable Set at time t from a given set $C \subset \mathbf{R}^n$ at time t_0, $t_0 \leq t$, is given by

$$\mathcal{A}(t; t_0, C) = \{x \in \mathbf{R}^n : V(t, x) \leq 0\}$$

where

$$\begin{aligned} V_t(t,x) + \mathcal{H}(t, x, V_x(t,x)) &= 0 \quad \forall (t,x) \in \mathbf{R}^+ \times S \\ V(t_0, x) &= {d_C}^2(x), \end{aligned} \tag{A.1}$$

where $S \subset \mathbf{R}^n$ is some domain of definition, $d_C(\cdot)$ is the usual distance function to set C, $\mathcal{H}(t, x, p) = \sup_{u \in \Omega} \{p \cdot f(t, x, u)\}$ is the Hamiltonian, $u(t) \in \Omega$ represent the control constraints, and $\dot{x} = f(t, x, u)$ are the system dynamics. Solutions to (A.1) in the classical (C^1) sense fail to exist in general, and some generalized concept is needed. Weaker solution concepts such as the viscosity for continuous solutions, (115, 138), and proximal normal for lower semi-continuous solutions (117) have been developed.

These references also include a characterization of Backward Attainable Sets, and (119) also targets the verification problem for moving targets specifically. Moreover, this reference includes comparison results enabling to replace the solution to the HJE by less computationally demanding upper and lower estimates of its solution which can be used for verification problems in reachability analysis.

An alternative characterization of Attainable Set in terms of a proximal normal solution to the HJE associated to a dynamic control system in the form of a differential inclusion appears in (116).

The complexity of its computation, or approximation, is strongly linked to the nature of the state transition map of the dynamic system. While, for discrete systems, there are already many results and tools for verification, (139), developments for continuum time and for the more recent hybrid systems, many challenges still remain. Unfortunately, it is, in general, very difficult to compute exactly the Attainable Set of systems evolving in continuum time. In fact, this is as difficult as to integrate the dynamics over time for all possible control strategies. It is no wonder, that many techniques have been addressed to investigate the properties of Attainable Sets and to define efficient ways of approximating them.

The emergence of the so-called level set methods, (120, 121), enabled the efficient computational approximation to viscosity solutions to the HJE and the associated convergence proofs of the numerical algorithms has been established. Although requiring somewhat more limiting assumptions, the ordered upwind methods, see (140), are highly efficient from the computational point of view. Thus, the numerical computation of Attainable Sets involve the definition of a level set function for the region with appropriate properties and, then, propagate it on the region of interest with the help of HJE. For more details on level set methods, check Ian Mitchell's level set methods webpage, http://www.cs.ubc.ca/$\sim$mitchell/ToolboxLS/index.html.

These methods have been applied in a wide range of applications. See, for example, (121, 141).

An alternative approach, exhibiting features similar to those of level set methods, has been provided by viability theory, see (142, 143), which has been used to treat a large number of applications.

A quite different approach consists in propagating ellipsoidal approximations to the Attainable Set for the case in which the underlying dynamics are linear (119, 131). This approach may involve either outer or inner approximations and has the great advantage of its very low complexity since only a very small number of parameters has to be propagated. It has also been shown that good accuracy can be achieved in the approximation of nonconvex backward Attainable Sets.

Appendix B

Polyhedral Approximations

In this appendix we describe an efficient recursive algorithm to generate both inner and outer polyhedral approximations to the Attainable Sets derived from the one in (133). Other approaches to approximate Attainable Sets have been considered, (125, 131, 135). However, key reasons for choosing this method are, on the one hand its simplicity, and, on the other hand, the fact that it yields affine constraints in the associated optimization problem for which there are very efficient solvers available. Moreover, the proposed scheme may produce either inner or outer approximations. While the former is of required to ensure feasibility, the later is needed to ensure safety type of properties. The general idea behind the approach for estimating the Attainable Set relies in the observation, e.g., (134), that, when the cost function depends linearly on the state variable at the final time only, it is known that the optimal value is reached at the boundary of the Attainable Set.

It is a classic result that if $u^*_{|[t_0,t_1]}$ (locally) minimizes $-\langle \alpha, x(t_1)\rangle$, then $x^*(t_1)$ is on the boundary of the Attainable Set $\mathcal{A}_f(t_1; t_0, x_0)$, and α is said to be normal to the Attainable Set at the $x^*(t_1)$, (126). In addition, the propagation of this relation holds for all intermediate values of time, in the sense that the adjoint variable $p(t)$, defined by $p(t_1) = \alpha$, and $-\dot{p}(t) \in \partial_x H(t, x^*(t), p(t))$, being $H(t, x, p) := \sup_{u\in\Omega}\{\langle p, f(t, x, u)\rangle$ the Hamiltonian associated with the considered dynamic control system, is normal to $\mathcal{A}_f(t; t_0, x_0)$ at $x^*(t)$. In a sense, $p(t)$ provides "locally" (i.e., in a neighborhood of time t) a quantitative indication of the sensitivity of the optimal control function and can be used to compute its value at any intermediate time value.

For a given $\alpha \in I\!R^n$ (assume $\|\alpha\| = 1$), denote by $x^*(t_1; \alpha)$, to be the optimal state trajectory at the final time, solution to the following optimal control problem:

$$(P(\alpha)) \text{ Minimize } \{-\langle \alpha, x(t_1) \rangle : (5.1), (5.2) \text{ hold on } [t_0, t_1]\}.$$

Since $x^*(t_1; \alpha)$ is a boundary point of the $\mathcal{A}_f(t_1; t_0, x_0)$, by varying α, different boundary points can be produced. These points can be regarded as vertices of the polyhedron approximating the Attainable Set. These considerations form the basis for an recursive algorithm generating a polyhedron, approximating the convex hull of the Attainable Set. To facilitate the exposition, we will consider the Attainable Set to be convex at this point. Although, the algorithm presented below produces an inner approximation to the Attainable Set, it is not difficult to change it in order to obtain an outer approximation.

1. Initialization.

   Specify the threshold for the facet error $\bar{\varepsilon} > 0$. By facet error, it is meant the maximum Euclidean distance from any point in the facet to the surface of the Attainable Set.

   It involves the following steps: (i) Computation of the set of initial facets. This is a minimal polyhedron, in the sense that it has only two opposite facets given as the convex hull of n vertices, and thus contained in a $n-1$ dimensional linear subspace. In $I\!R^n$, it can be computed as the convex hull of n vertices constituting a simplex in $I\!R^{n-1}$, each one obtained by solving an optimal control problem $P(\alpha)$ for appropriate choices of the vector α, (133); (ii) Initialize counters of: (a) iterations $k = 0$, (b) vertices - $V_k = n$, and (c) facets - $F_k = 2$.

2. Detection of unsatisfactory facets.

   For each new facet $\mathcal{F}_j$, $j = F_k - n + 1, \ldots, F_k$, compute the respective error ε_j, given by the maximum distance between the facet and subset of the boundary of the Attainable Set with the shortest projection distance on the facet. This is done in two stages: (i) Compute the point $x_j^*(t_1)$ on the boundary of the Attainable Set by solving $(P(\alpha))$ with $\alpha = f_j$ being f_j an unit vector orthogonal to the facet F_j and point outwards w.r.t. the polyhedron; and (ii) Let $\varepsilon_j = \|x_j^*(t_1) - \pi_j\|$ where π_j is the projection of $x_j^*(t_1)$ on F_j. The set of unsatisfactory facets is given by

$UF := \{F_j : \varepsilon_j > \bar{\varepsilon}\}$ in which all the items with error greater than $\bar{\varepsilon}$ are ordered in an increasing order of the associated error ε_j.

Once this set becomes empty, then the algorithm stops and the inner polyhedral approximation to the Attainable Set will be given by the convex hull of all vertices, i.e.,

$$\bar{co}\mathcal{A}_f^{\bar{N}}(t_1; t_0, x_0) = \bar{co}\{x_j^*(t_1) : j = 1, \ldots, \bar{N}\},$$

where $\bar{N} = V_{\bar{k}}$, and $\bar{k}$ is the number of iterations needed. In case the Attainable Set is convex, then we have $\mathcal{A}_f^{\bar{N}}(t_1; t_0, x_0) = \bar{co}\mathcal{A}_f^{\bar{N}}(t_1; t_0, x_0)$.

3. Computation of new facets.

   While $UF \neq \emptyset$, replace the last facet in UF, $\mathcal{F}_{F_k}$ by n facets, each one obtained as the convex hull of $x_{F_k}^*(t_1)$ and each pair of adjoining vertices of $\mathcal{F}_{F_k}$.

4. Update counters Let $k = k + 1$, $V_k = V_{k-1} + 1$, and $F_k = F_{k-1} + n - 1$, and go to step 2.

Clearly, the number of vertices, and, thus, of faces of the polyhedron is, in this algorithm, increases successively until the approximation error, given by the worst case facet error is below some predefined upper bound $\bar{\varepsilon}$.

In order to get a more accurate nonconvex polyhedral approximation to nonconvex Attainable Sets, an additional procedure was developed in (133). It involves two stages. The first one, consisting in the detection of a region of non-strict convexity, may encompass two cases: (i) points of the boundary of the approximating polyhedron which are not in the Attainable Set, and (ii) points in of the approximating polyhedron which are not in the Attainable Set. The second stage involves the generation of a local approximation of the boundary of the Attainable Set. The construction underlying this algorithm allow us to obtain the following result on the estimation of the approximation error and in the degree of sub-optimality of the corresponding optimal control problem. For this we need the notion of Hausdorff distance d_H between two sets. Given A, $B \subset I\!R^n$ and $d(d, C) = \inf\{\|c - d\| : c \in C\}$,

$$d_H(A, B) = \max\left\{\max_{x \in A}\{d(x, B)\}, \max_{y \in B}\{d(y, A)\}\right\}.$$

Proposition B.0.1 *Given a time horizon Δ, $\forall \varepsilon > 0$, $\exists N(\varepsilon, \Delta) \in \mathbb{N}$ such that, $\forall N \geq N(\varepsilon, \Delta)$,*

$$d_H\left(\bar{co}\mathcal{A}_f^N(t+\Delta; t, x(t)), \bar{co}\mathcal{A}_f(t+\Delta; t, x(t))\right) < \varepsilon.$$

If the Attainable Set is convex, then $\bar{co}$ may be suppressed. Moreover, under some mild assumptions, there is some constant $\tilde{K}$ related to the Lipschitz constant of the system dynamics such that

$$Inf(P_\Delta^N) \leq Inf(P_\Delta) + \tilde{K} d_H(\mathcal{A}_f^N(\cdot), \mathcal{A}_f(\cdot)),$$

where $\mathcal{A}_f^N$ is a N facet inner polyhedral approximation to $\mathcal{A}_f$ and both are evaluated at $(t+\Delta; t, x(t))$, and (P_Δ^N), (P_Δ) are the corresponding optimization problems on the time interval $[t, t+\Delta]$ with the initial state $x(t)$.

This property allows us to estimate how suboptimal the solution to the optimization problem is from the one obtained when the exact Attainable Set is used as well as how far the solutions from each one of the problems are from each other. An analogue result can be obtained for the case in which the outer approximation is considered.

Now, we are ready to present the basic approximated Attainable Set MPC scheme:

1. Initialization.

2. Compute $\mathcal{A}_f^N(t+\Delta; t, x(t))$.

3. Compute $z^* = \underset{z \in \mathcal{A}_f^N(t+\Delta; t, x(t))}{\operatorname{argmin}} \{V(t+\Delta, z)\}$.
   Compute u^* on $[t, t+\Delta]$ so that $x(t+\Delta) = z^*$.

4. Apply u^* during $[t, t+\Delta]$.

5. Sample x at $t+\Delta$ to obtain $\bar{x} = x(t+\Delta)$.

6. Slide time, i.e., $t = t+\Delta$, let $x(t) = \bar{x}$, and goto 2..

A number of observations are in order. This scheme has the advantage of combining the long term optimization perspective encapsulated in the Value Function and Attainable Set estimates with a low computational burden inherent to the fact that (i) the optimization carried out in each iteration is only in the short term horizon, and (ii) the Value Function and the Attainable Set approximations can be computed off-line and obtained for the current position in the state space via a look-up table.

In a dynamic world in which obstacles may emerge or other changes in the environment detectable by the on-board sensors at the execution time may occur, both the Attainable Set and the Value Function will have to be adapted. Here, two interesting observations in what concerns the emergence of obstacles, which are particularly important from the computational point of view are (i) since it is propagated backwards, the Value Function can be adapted only from the new boundary condition detected within the detection horizon until the current moment, and (ii) the new feasible set to be considered as constraint set can be easily obtained as the intersection of the previews feasible set with the new free space dictated by the detected obstacles. This is particularly useful for formation control problems since the specification of the formation can be cast in terms of phase constraints.

Appendix C

Practical simple AUV model identification procedures

In this appendix we discuss the experimental approach and the procedures in order to identify the models of the various equations of motion for the AUVs in LSTS. This discussion involves three main components:

- Modeling of the various modes of operation
- Description of the identification methods
- Identification data gathering

As an example we present in Figure C.1 one of the AUVs developed at LSTS that has been used for modeling. LAUV is a small (110x16 cm) yet modular autonomous low-cost submarine which can be used for different types of operations depending on payload configuration. LAUV vehicles provide a maximum operating depth of 50m and 1.5 m/s nominal speeds for oceanographic and environmental surveys. It is equipped with one propeller and 4 actuated fins, and the main payload is listed in Table C.1. The energy provided by a set of rechargeable Lithium-Ion batteries lasts for over 8 hours at the nominal speed. The standard configuration of the sensor payload includes a Conductivity, Temperature and Depth (CTD) sensor. The onboard navigation suite includes a low-cost inertial measurement unit, a depth sensor, a GPS unit and a LBL system for acoustic positioning that is used when the vehicle is underwater and thus GPS-restrained. LAUV also uses a WiFi and GSM for communications at the surface.

Figure C.1: LAUV Package: Vehicle, LBL beacons, PAL (Portable Acoustic Locater), and battery charger

Figure C.2: APDL: Porto local harbor

Onboard software is based on DUNE (DUNE: Uniform Navigational Environment), which provides a modular architecture for supporting sensors, actuators and the creation of controllers using the concept of messaging among asynchronous tasks.

The mission site used for testing ongoing developments is provided by APDL (Administração dos Portos do Douro, Leixões e Viana do Castelo), the local Porto harbour. which can be seen on figure C.2. This location is very useful since it is on the coast, providing access to open sea and, on the other hand, provides a secure enclosed area for rapid tests using sea water.

C.1 Modeling of the various modes of operation

We consider the modes of operation: Surge, Yaw, Pitch, and Heave and, for each one of them, a model will be presented next.

Item	**Description**
Batteries	SAFT Li-Ion - 25,2V at 5.8Ah (x2)
IMU	3DM-GX1
CTD	Mark & Wedell
Sidescan	Marine Sonics HDS - 900kHz
DVL/ADCP	LinkQuest
Wifi	MiniStation2
GPS	EVK-5H
GSM/GPRS	Telit GM862
Altimeter	Imagenex 852
Pinger	Imagenex 852
Main CPU	Sonotronics
Auxiliary CPU	PC104 ARM
Pressure Sensor	Gems
Forward Sonar	Imagenex 852
Acoustic Board	ULST Custom Made
Leak Sensor	ULST Custom Made

Table C.1: LAUV AUV main equipment

Surge Model.

The resulting forces in the longitudinal direction, considering surge motion only, is equal to the thruster's force and the longitudinal drag force. The equation of motion is given by

$$(m - X_{\dot{u}})\dot{u} = X_{u|u|}u|u| + F_{prop} \tag{C.1}$$

where m is the mass, $X_{\dot{u}}$ is the added mass coefficient, $X_{u|u|}$ is the quadratic drag coefficient and F_{prop} is the thruster force. These terms can be difficult to obtain for irregular shapes. A recommended approach would be to approximate the vehicle's shape by a prolate ellipsoid[1] as depicted in figure C.3. For that, and since the ellipsoid's shape will not perfectly match the real vehicle's, the radius $b = c$ should be matched with that of the AUV and the length a should be adjusted so that the vehicle's volume matches that of the ellipsoid.

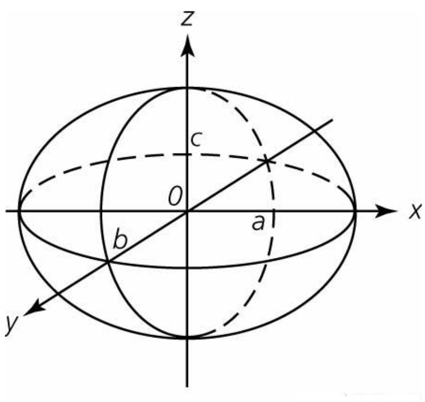


Figure C.3: Three-dimensional ellipsoid

By using these values, the coefficient $X_{\dot{u}}$ can be computed as:

$$X_{\dot{u}} = -k_x \frac{B}{g}$$

where k_x can be found by using Table C.2 and B/g is the neutrally buoyant mass

$$\frac{B}{g} = \frac{4}{3}\pi abc\rho.$$

The term ρ is the fluid's density. The coefficient $X_{u|u|}$ can be found with

$$X_{u|u|} = \frac{1}{2}C_D A\rho$$

[1] An ellipsoid where $a > b, c$ and $b = c$.

where C_D is the drag coefficient for the vehicles's cross-sectional area A. For an ellipsoid with $b = c$, then $A = \pi b^2$. According to (144), the value for C_D should be around 0.2 but we will define it as a parameter to be identified.

The force exerted by the thruster is given by

$$F_{prop} = K_T \rho D^4 n|n|$$

where D is the propeller's diameter, n the propeller revolutions and K_T is constant and related to the speed of advance:

$$K_T = K_T\left(\frac{V_A}{nD}\right) = K_T(J_0)$$

where V_A is the speed of advance in m/s and J_0 is the non-dimensional speed of advance. The value for K_T should be between 0.1 and 0.4 but we will treat this parameter as one to be identified. We can also observe if it lays within this interval as a measure of the parameters identification congruency.

In fact F_{prop} contains another term given by $\gamma_0 J \rho D^4 n|n|$, where γ_0 is the $K_T(J)$ function's slope for a certain operating point. It represents a thrust reduction. We will not explicitly take this term into account, since it is quite difficult to tell this term's effect apart from the effect of the first term, $K_T \rho D^4 n|n|$.

In order to find the values of coefficients C_D and K_T, the procedures in Section C.2 should be followed to compute α_u and β_u of the following equation

$$\dot{u} = \alpha_u u|u| + \beta_u n|n|$$

C_D and K_T comes from the following relations

$$C_D = 2\frac{\alpha_u(m - X_{\dot{u}})}{A\rho} \quad K_T = \frac{\beta_u(m - X_{\dot{u}})}{\rho D^4}$$

Yaw Model.

Sideslip is the skidding motion presented by the vehicle when it moves with both in surge and sway. There are two possible Yaw equations of motion depending if we expect from the vehicle sideslip behavior or not. At this stage we do not know which model fits better and therefore both will be present. Here, we consider two cases

a/b	k_x	k_y	k_r
1	0.5	0.5	0
1.5	0.305	0.621	0.094
2	0.209	0.702	0.24
2.51	0.156	0.763	0.367
2.99	0.122	0.803	0.465
3.99	0.082	0.86	0.608
4.99	0.059	0.895	0.701
6.01	0.045	0.918	0.764
6.97	0.036	0.933	0.805
8.01	0.029	0.945	0.84
9.02	0.024	0.954	0.865
9.97	0.021	0.96	0.883

Table C.2: Added mass k coefficient table

- Neglecting side-slip The nonlinear equation for the vehicle's turning rate can be written as

$$(I_{zz} - N_{\dot{r}})\dot{r} = N_r r + N_\delta \delta_r \tag{C.2}$$

where I_{zz} is the mass moment of inertia along the z axis, $N_{\dot{r}}$ is the added mass coefficient, N_r is the linear damping coefficient and N_δ is the rudder fins' lift coefficient. Similarly to the Surge model above, if the shape of the AUV is approximated by a prolate ellipsoid, I_{zz} and $N_{\dot{r}}$ can be calculated as:

$$I_{zz} = \frac{4(a^2+b^2)\pi abc\rho}{15} \quad N_{\dot{r}} = k_r I_{zz}$$

where k_r can be taken from Table C.2. Coefficient N_r has to be identified using methods suggested in this document and the rudder fins' lift coefficient is given by

$$N_\delta = \frac{1}{2} x_{fin} \frac{\partial C_f}{\partial \alpha_f} A_f u|u|$$

where x_{fin} is the fins' x position relative to the center of gravity, A_f is the fins' face area and C_f is a coefficient that comes as a function of the angle of attack on the fin, $C_f = C_f(\alpha_f)$. This term should be identified, as it can be difficult to obtain any other way. We will also assume that its partial derivative with respect to alpha is constant. In order to find the values of coefficients N_r and

C_f, the procedures in Section C.2 should be followed to compute α_r and β_r and then following relations should be used. In particular, if we define the following equation for the parameter identification,

$$\dot{r} = \alpha_r r + \beta_r u|u|$$

we get

$$N_r = \alpha_r (I_{zz} - N_{\dot{r}}) \quad \frac{\partial C_f}{\partial \alpha_f} = 2\frac{\beta_r (I_{zz} - N_{\dot{r}})}{x_{fin} A_f}$$

- Including side-slip If we do not neglect the side-slip, the equation for turning rate comes:

$$(I_{zz} - N_{\dot{r}})\dot{r} = N_r r + N_v v + N_\delta \delta_r \tag{C.3}$$

with I_{zz}, $N_{\dot{r}}$ and N_δ defined above. N_v is another parameter to be identified.

We are introducing this scenario in order to evaluate and compare how good these models are. The performance index to be used is given by:

$$J_i = \frac{\sqrt{\frac{\sum_{k=1}^{N}(y_k^{real} - y_k^{model})^2}{N}}}{\mathrm{RMS}(y_k^{real})} = \frac{\sqrt{\frac{\sum_{k=1}^{N}(y_k^{real} - y_k^{model})^2}{N}}}{\sqrt{\frac{\sum_{k=1}^{N} y_k^{real2}}{N}}} \tag{C.4}$$

Pitch Model.

Assuming the vehicle has only surge, heave and pitch speeds, the nonlinear model can be written as:

$$(I_{yy} - M_{\dot{q}})\dot{q} = (z_B B - z_G W)\sin\theta + M_q q + M_w w + M_\delta \delta_s \tag{C.5}$$

where I_{yy} is the moment of inertia along the y axis, $M_{\dot{q}}$ is the added mass coefficient, z_B is the z coordinate of the center of buoyancy, B is the buoyancy force, z_G is the z coordinate of the center of gravity, W is the weight of the vehicle, M_q and M_w are the drag coefficients for, respectively, pitch and heave motion, and M_δ is the fins' lift coefficient.

Equation (C.5) can be simplified with a few assumptions. The term z_B is assumed to be zero, as it can be defined as coincident with the body-fixed frame's origin. The trigonometric function $\sin\theta$ can be approximated by θ (in radians) for small angles ($\theta \leq \frac{\pi}{6}$). The remaining terms either have to be calculated or identified. We can now rewrite equation (C.5) as:

$$(I_{yy} - M_{\dot{q}})\dot{q} = -z_G W\theta + M_q q + M_w w + M_\delta \delta_s \tag{C.6}$$

Again, if we approximate the shape of the vehicle as a prolate ellipsoid, I_{yy} and $M_{\dot{q}}$ can be computed as

$$I_{yy} = \frac{4(a^2+c^2)\pi abc\rho}{15} \quad M_{\dot{q}} = k_r I_{yy}$$

where k_r can be taken from Table C.2. The vehicle's weight can be computed as $W = mg$. Terms z_G, M_q, M_w and M_δ must be identified. Similarly to the Yaw model described above:

$$M_\delta = \frac{1}{2} x_{fin} \frac{\partial C_f}{\partial \alpha_f} A_f u|u|$$

If we run the parameter identification method with 4 terms

$$\dot{q} = \alpha_q \theta + \beta_q q + \gamma_q w + \epsilon_q u|u|$$

we get

$$z_G = -\frac{\alpha_q(I_{yy} - M_{\dot{q}})}{W} \quad M_q = \beta_q(I_{yy} - M_{\dot{q}})$$

$$M_w = \gamma_q(I_{yy} - M_{\dot{q}}) \quad \frac{\partial C_f}{\partial \alpha_f} = 2\frac{\epsilon_q(I_{yy} - M_{\dot{q}})}{x_{fin} A_f}$$

Heave Model.

Assuming again only surge, heave and pitch motions, the nonlinear model can be written as

$$(m - Z_{\dot{w}})\dot{w} = (W - B)\cos\theta + mU_0 q + Z_q q + Z_w w + Z_\delta \delta_s \tag{C.7}$$

where $Z_{\dot{w}}$ is the vehicle's added mass coefficient, U_0 is the vehicle's rated surge speed, Z_q and Z_w are the linear damping coefficients about axis y and along z respectively, and Z_δ is the fin's lift coefficient. We will assume a neutrally boyant vehicle, that is, $W = B$, and therefore, the whole first term will be null. Coefficient $Z_{\dot{w}}$ can be calculated as $Z_{\dot{w}} = -k_z m$ where $k_z = k_y$ so it can be taken from Table C.2.

The simplified model can now be written as

$$(m - Z_{\dot{w}})\dot{w} = mU_0 q + Z_q q + Z_w w + Z_\delta \delta_s \tag{C.8}$$

The speed U_0 can be identified or taken from the mission data, Z_q and Z_w must be identified and Z_δ is given by

$$Z_\delta = \frac{1}{2}\frac{\partial C_f}{\partial \alpha_f} A_f u|u| = \frac{M_\delta}{x_{fin}}.$$

As occurred in the Pitch model above, the term $\frac{\partial C_f}{\partial \alpha_f}$ has to be identified. By using the parameters identification method with four terms as in

$$\dot{w} = \alpha_w q + \beta_w q + \gamma_w w + \epsilon_w u|u|$$

we will obtain

$$U_0 = \frac{\alpha_w(m - Z_{\dot{w}})}{m} \quad Z_q = \beta_w(m - Z_{\dot{w}}),$$
$$Z_w = \gamma_w(m - Z_{\dot{w}}) \quad \frac{\partial C_f}{\partial \alpha_f} = 2\frac{\epsilon_w(m - Z_{\dot{w}})}{A_f}.$$

Final model.

We can finally resume the complete set of equations for the simplified AUV model as follows. This will be of great importance for the formation control design and simulation.

$$\dot{\nu} = \begin{bmatrix} \frac{X_{u|u|}u|u|+F_{prop}}{(m-X_{\dot{u}})} \\ \frac{mU_0q+Z_qq+Z_ww+Z_\delta\delta_s}{(m-Z_{\dot{w}})} \\ \frac{-z_GW\theta+M_qq+M_ww+M_\delta\delta_s}{(I_{yy}-M_{\dot{q}})} \\ \frac{N_rr+N_vv+N_\delta\delta_r}{(I_{zz}-N_{\dot{r}})} \end{bmatrix} \tag{C.9}$$

$$\dot{\eta} = \begin{bmatrix} u\cos(\psi) \\ u\sin(\psi) \\ w \\ q \\ r \end{bmatrix} \tag{C.10}$$

where $\nu = [u, w, q, r]^T$ and $\eta = [x, y, z, \theta, \psi]^T$.

C.2 Identification methods

System identification is the process of modeling systems where both model parameters and equations are unknown (145). The procedure encompasses measuring the system's inputs and outputs and try to determine a mathematical relation between them without going into the details of what is actually happening inside the system. Typical types of models used in System Identification range from black-box to white-box. Black-box models include no prior information about the system while White-box are the ones where physical laws (e.g. Newton) describe perfectly the system behavior. Grey-box models obviously stand in between, where, some already known information is included, and the remaining parameters are estimated. In this section we propose a method for experimental identification of the parameters, for the AUV decoupled equations of motion on surge, heave, pitch and yaw. This can be done by using least squares (146) and Kalman filter (144). Moreover, input sequences will be specified so that data read from sensors is used in the context of parameters estimation.

In order to identify some of the parameters for the AUV model equations we use alpha-beta parameter identification technique. The alpha-beta parameter technique uses least square methods to produce the estimate.

Alpha and Beta Parameters Identification

We select a model of the input/output response of the general form

$$y(t) = H^T(t)\theta(t) + v(t) \tag{C.11}$$

where $\theta(t) \in \mathbb{R}^{n+m}$ is a parameter vector that is ideally constant where m is the number of input measurements and n is the number of output measurements, the matrix $H(t) \in \mathbb{R}^{n+m}$ contains all inputs/output measurements, $y(t) \in \mathbb{R}$ is the primary output and $v(t) \in \mathbb{R}$ is assumed to be a zero mean white gaussian noise signal.

Discretizing we may write the time difference equation as

$$y_t - a_1 y_{t-1} - a_2 y_{t-2} - \ldots - a_n y_{t-n} = b_1 u_{t-1} + b_2 u_{t-2} + \ldots + b_m u_{t-m} \tag{C.12}$$

or

$$y_t = \underbrace{[y_{t-1}, y_{t-2}, \ldots, y_{t-n}, u_{t-1}, u_{t-2}, \ldots, u_{t-m}]}_{H^T(t)} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}}_{\theta} \tag{C.13}$$

which can be translated into a discrete time transfer function as:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{\displaystyle\sum_{i=1}^{m} b_i z^{-i}}{1 - \displaystyle\sum_{j=1}^{n} a_j z^{-j}} \tag{C.14}$$

where z^{-i} is the delay operator in discrete time.

Model with Alpha and Beta Parameters

Critical to the solution is the use of an adequate model of the input/output response of the system. Many models with several different parameters are possible but in this work we propose the use of the previous derived model.

$$\begin{aligned}
\dot{u}(t) &= \alpha_u u(t)|u(t)| + \beta_u n(t)|n(t)| + v_u(t) &\quad (C.15)\\
\dot{w}(t) &= \alpha_w w(t) + \beta_w q(t) + \gamma_w \delta_s(t) + v_w(t) &\quad (C.16)\\
\dot{q}(t) &= \alpha_q \theta(t) + \beta_q q(t) + \gamma_q w(t) + \epsilon_q \delta_s(t) + v_q(t) &\quad (C.17)\\
\dot{r}(t) &= \alpha_r r(t) + \beta_r v(t) + \gamma_r \delta_r(t) + v_r(t) &\quad (C.18)
\end{aligned}$$

where u, v, w, q and r represent the vehicle's surge, sway, heave, pitch and yaw velocities and δ_s and δ_r are respectively stern and rudder actuator fins. Sway equation of motion was neglected due to lack of relevance for our AUVs. However, sideslip motion is considered.

The purpose of the identification is to find the equations of motion parameter sets (α_u, β_u), $(\alpha_w, \beta_w, \gamma_w)$, $(\alpha_q, \beta_q, \gamma_q, \epsilon_q)$ and $(\alpha_r, \beta_r, \gamma_r)$. Next we will show how these parameters can be obtained using two different methods.

Solution Using Least Squares

We seek a solution $\hat{\theta}(t)$ as an estimate of $\theta(t)$ such that the error $\tilde{\theta}(t) = \theta(t) - \hat{\theta}(t)$ can be shown to decrease. The true parameters $\theta(t)$ are not available so the error to minimize is expressed as the error of equation (C.11) and is given by

$$e(t) = y(t) - H^T(t)\hat{\theta}(t) \quad (C.19)$$

With least squares, we try to find a $\hat{\theta}(t)$, that will minimize the sum of errors, so that over some time interval, the effects of noise are canceled. If we define the scalar positive squared error measure as $J(n) = \frac{1}{2}\sum_{t=1}^{n} e^T(t)e(t)$, then the minimization of J is given by

$$\frac{dJ}{d\hat{\theta}} = 0 = -\sum_{t=1}^{n} H^T(t)e(t) \quad (C.20)$$

yielding

$$0 = -\sum_{t=1}^{n} H^T(t)(y(t) - H(t)\hat{\theta}(t)) \quad \Rightarrow \quad \sum_{t=1}^{n} H^T(t)y(t) = \sum_{t=1}^{n} H^T(t)H(t)\hat{\theta}(t) \quad \text{(C.21)}$$

The term $\hat{\theta}(n)$ can then be found with:

$$\hat{\theta}(n) = \frac{\sum_{t=1}^{n} H^T(t)y(t)}{\sum_{t=1}^{n} H^T(t)H(t)} \quad \text{(C.22)}$$

Equation (C.22) is used for "batch" processing where the block size is n. This is sometimes very useful and a moving batch window can be used to produce a running estimate of the most recent estimates of parameters.

Solution Using Kalman Filter

Similarly to the least squares methods, a Kalman filter methodology can be applied to model parameter identification. Assuming, once again, that the parameters are constant, we know that

$$\begin{aligned} \theta(k) &= \theta(k-1) \\ y(k) &= H(k)\theta(k) + v(k) \end{aligned}$$

where $v(k)$ is a white noise signal with variance σ_r^2. In order to apply this technique to a model such as the one in equation (C.15), we must write it in the discrete time form:

$$\frac{u(k+1) - u(k)}{T_s} = \alpha u(k)|u(k)| + \beta n(k)|n(k)| + v(k)$$

where $v(k)$ is the white noise signal with variance σ_r^2. Now let

$$\hat{\theta} = \begin{bmatrix} \hat{\alpha}(k) \\ \hat{\beta}(k) \\ 1 \end{bmatrix} \quad \text{and} \quad H(k) = [u(k)|u(k)|T_s,\ n(k)|n(k)|T_s,\ u(k)] \quad \text{(C.23)}$$

so that this way we can write the output data as a function of the estimated parameters

$$u(k+1) = H(k)\hat{\theta}(k) + v(k)$$

The Kalman filter methodology assumes

$$\hat{\theta}(k+1) = \hat{\theta}(k) + q(k)$$

where $q(k)$ is a white noise signal (process noise) with variance σ_q^2, which could be defined as zero, since we are assuming that the parameters are constant. The Kalman filter algorithm then comes as:

$$\begin{aligned} P(k) &= AP(k)A^T + Q \\ K(k) &= P(k)H^T(k)[H(k)P(k)H^T(k) + R]^{-1} \\ \hat{\theta}(k+1) &= \hat{\theta}(k) + K(k)(u(k+1) - H(k)\hat{\theta}(k)) \\ P(k+1) &= [I - K(k)H(k)]P(k) \end{aligned}$$

where A is the linear model matrix, Q is a diagonal matrix whose terms are equal to the variance of each parameter, R is also diagonal with measurement noise's variance as coefficients, K is the Kalman filter gain matrix and P is the error covariance matrix. Matrix P must be properly initialized, by using each term in its diagonal as a weight representing the uncertainty of our first estimated set of parameters $\hat{\theta}(0)$. For instance, knowing that the third parameter will always be 1, we should initialize the third term in the diagonal of $P(0)$ as 0, meaning we have full certainty on that value.

Unlike the least squares method, this technique does not use the whole scope of data at once for estimating the parameters. This methodology, as described above, can either be used online or off-line, while the least squares may only be used off-line. Note that using the Kalman filter off-line will present no advantage whatsoever over the use of the least squares method. As a matter of fact, we will advise against the use of the Kalman filter for off-line identification, since improper initialization of the matrices P, Q and R may never yield satisfying results, a risk that we don't have to take by using least squares.

Next will discuss how to get data from the vehicle to identify the parameters.

C.3 Gathering data for identification

Acquiring good data for the identification process is critical. The way data is obtained determines the success of the parameters identification. Moreover, the existence of inverse of the $\sum_{t=1}^{n} H^T(t)H(t)$ in equation (C.22) depends on the quality of the information contained in the gathered data.

There are also other issues that one might want to avoid like performing the data gathering in the presence of currents or near the surface or bottom. These introduce significant disturbances leading to an incorrect model. We chose the APDL (Porto Local Harbor Authority) port to be the operation site, being the data gathered below a depth of 2 meters to avoid wave effects and 2 meters above the bottom to avoid both "boundary" effects as well as potential collisions.

The most important condition to design a good vehicle input sequence is to do it in a way that it's frequency response is similar to the system frequency response. This way all system modes are correctly excited and the dynamics properly captured. Two different signals are proposed. The sum of 5 sines with different frequencies in the vehicle's bandwidth w_b or square wave (C.24) and a square signal (C.25).

$$i(t) = \sum_{n=1}^{5} sin(w_b/n) \tag{C.24}$$

$$i(t) = \begin{cases} v_{max}, 2kT_f < t < (2k+1)T_f \\ v_{min}, (2k+1)T_f < t < (2k+2)T_f, k \in \mathbb{N}_0^+ \end{cases} \tag{C.25}$$

where T_f is the time after which the system reaches v_{max} or v_{min} plus another 100% to let the system stabilize. For instance, if we are identifying the surge model, means that the signal to the thruster toggles between v_{max} and v_{min} every $T_f = 1.5\tau_u$, where τ_u is the first order approximation time constant for the surge model.

Even though the sine signal makes sense from the frequency response point of view, in practice this signal is not very easy to implement in the vehicle due to actuators saturation. Square signals will be used instead. The length of the input vector should be big enough to allow both the identification and model validation, typically one half each.

For the sake of simplicity we illustrate only the method to obtain the parameters for the surge equation of motion. A similar approach must be followed to obtain the remaining parameters.

Taking equation (C.15) and converting to discrete using Euler we get:

$$u(k+1) = u(k) + \alpha_u u(k)|u(k)|T_s + \beta_u n(k)|n(k)|T_s \tag{C.26}$$

where T_s is the discretization time step. Or in matrix representation:

$$u(k+1) - u(k) = \underbrace{\left[\begin{array}{cc} u(k)|u(k)|T_s & n(k)|n(k)|T_s \end{array}\right]}_{H(k)} \underbrace{\left[\begin{array}{c} \alpha_u \\ \beta_u \end{array}\right]}_{\theta} \tag{C.27}$$

which, apart from the noise v, is the equation (C.11) that we began with. Also note that we have moved the term $u(k)$ to the left side of the equation so that the third parameter in $\hat{\theta}$ is forced to be 1.

$$\underbrace{\left[\begin{array}{c} u(1) - u(0) \\ \vdots \\ u(p) - u(p-1) \end{array}\right]}_{Y} = \underbrace{\left[\begin{array}{cc} u(0)|u(0)|T_s & n(0)|n(0)|T_s \\ \vdots & \vdots \\ u(p-1)|u(p-1)|T_s & n(p-1)|n(p-1)|T_s \end{array}\right]}_{H} \underbrace{\left[\begin{array}{c} \alpha_u \\ \beta_u \end{array}\right]}_{\theta} \tag{C.28}$$

where $Y \in \mathcal{R}^p$ is a vector of all p measurements, $H \in \mathcal{R}^{p\times 2}$ the matrix of all inputs and outputs, and $\theta \in \mathcal{R}^2$ the vector of the parameters.

If we now use Matlab back-slash operator "\", we can compute the least squares as $\theta = H\backslash Y$, or we can use the Kalman filter approach. To build H and Y we need input/output data. From a previous AUV mission data set, a first order approximation time constant of $\tau_u = 10s$ was found. We choose $T_f = 20s$ and apply a square wave to the system and record it's output. To guarantee that the vehicle does not lose controllability, the propeller revolutions should not decrease below 50%. We think that the input sequence should range between $v_{min} = 50\%$ and $v_{max} = 100\%$.

The final procedure to identify surge model include:

1. Generate a mission with constant depth $@z = 3m$ with propeller speed toggling between 50% and 100% every $T_f = 20s$. The desired trajectory should be a set of GoTo maneuvers maintaining a straight line.

2. Execute. Record data with length L.

3. Use half of the recorded data to compute α_u and β_u using least squares/Kalman filter method.

4. Test the discovered model with the other half of the recorded vector.

If we use the data obtained from the vehicle represented in C.4 we obtain the following model.

$$u(k+1) - u(k) = -0.0364u(k)|u(k)|T_s + 0.0100n(k)|n(k)|T_s \tag{C.29}$$

Figure C.5 shows the data output of the identified model in comparison with the real data from the mission logs.

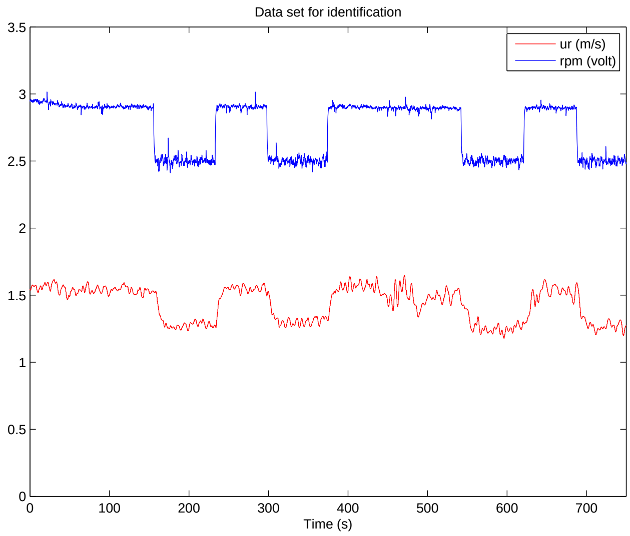


Figure C.4: Selected data set for identification: surge speed (red) and propeller revolutions (blue)

The same procedure can now be applied to the other equations of motion otaining the following parameters listed in Table C.3.

Parameter	Value	Parameter	Value
α_u	0.0364	α_r	-0.0220
β_u	0.0100	β_r	-0.1149
α_w	-0.0850	γ_q	0.0880
β_w	-0.0066	ϵ_q	-0.1198
γ_w	0.0041	α_q	-0.0184
		β_q	-0.7227

Table C.3: Final model identified parameters

C.4 Mission planning to generate identification data

In order to plan AUV missions, we used NEPTUS software (147, 148), https://lsts.fe.up.pt/toolchain/neptus. NEPTUS is a command, control, communication and information software infrastructure for the coordination and control of teams of multiple autonomous and semi-autonomous vehicles. It allows mission planning, supervision, and post-mission analysis. Using this framework, short AUV missions were planned and executed in order to collect the necessary data for identification.

For each model, different types of trajectories were defined so the right model parameters could be identified. These trajectories are defined as a composition of straight lines defined through a series of 3D waypoints. These trajectories are known as GoTo maneuvers where the vehicle travels through each waypoint. Next, we will describe just the yaw and pitch mission plans required for modeling.

Surge Model Identification Plan

Figure C.6 shows the plan adopted for the AUV to identify the Surge parameters. The waypoints were placed in a straight line. The vehicle has to increase and reduce it's speed at every waypoint. Every two waypoints the distance between each waypoint decreases. The starting distance is of 10 meters, then 8, 7, 6 and finally 5 meters apart. This way more frequency modes are probed in the identification process.

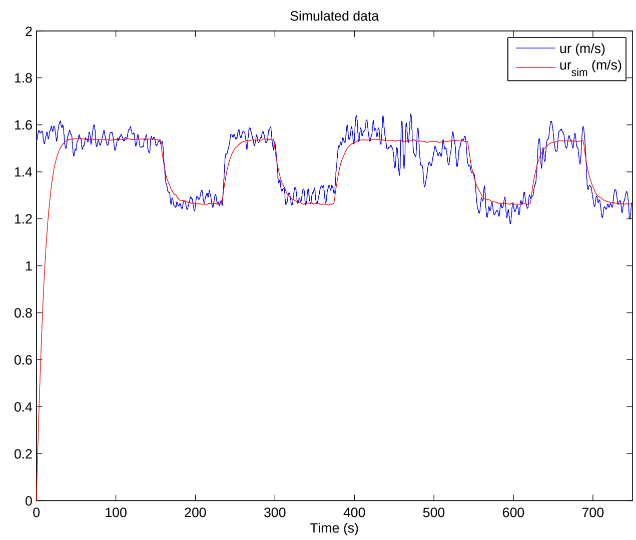


Figure C.5: Real and simulated data using the identified parameters α_u, β_u

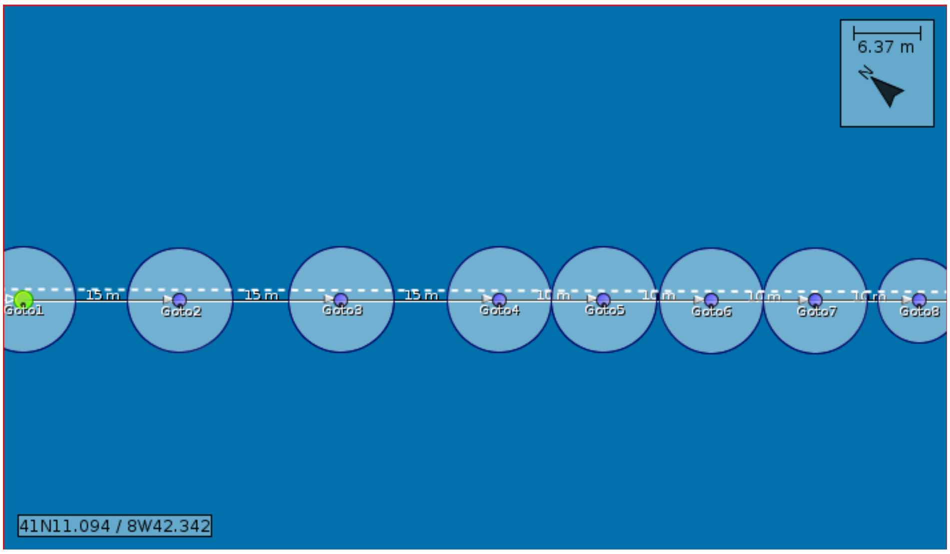


Figure C.6: Neptus mission plan for the AUV's surge model identification

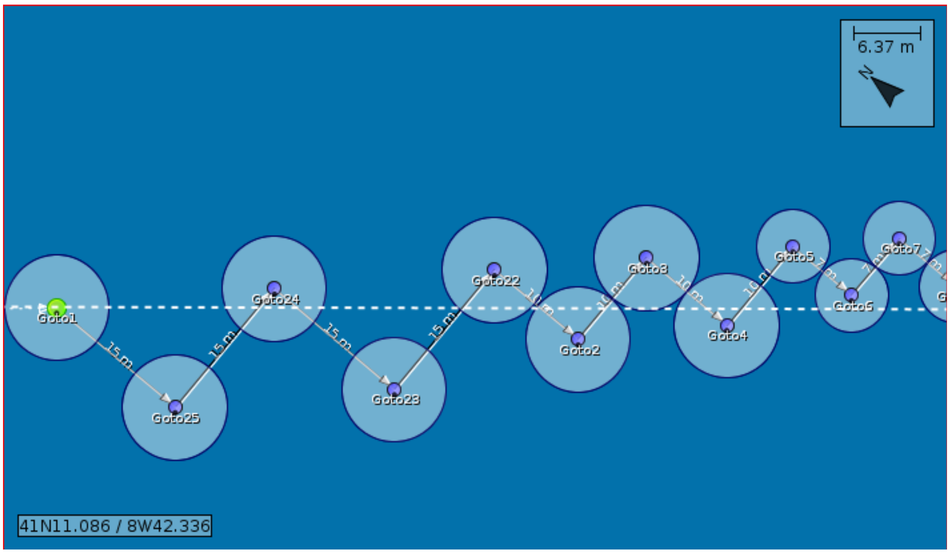


Figure C.7: Neptus mission plan for the AUV's yaw model identification

Yaw Model Identification Plan

Figure C.7 shows part of the steering plan adopted for the AUV. The waypoints were placed such that the vehicle has to take 90 degree turns in the horizontal plane at every waypoint. Every four waypoints the distance between each one of them decreases. We started with 15 meters, then 10, 7, 5, 4 and then 3 meters of distance between waypoints. This plan will swing the vehicle in the horizontal plane as shown in Figure C.7.