Coordination of waterborne AGVs

Zheng, Huarong

DOI
10.4233/uuid:f6aefbb0-1b95-44e9-a4dc-8e6c02d94f37

Publication date
2016

Document Version
Final published version

Citation (APA)
Zheng, H. (2016). *Coordination of waterborne AGVs*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:f6aefbb0-1b95-44e9-a4dc-8e6c02d94f37

Coordination of Waterborne AGVs

Huarong Zheng

Coordination of Waterborne AGVs

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
maandag 12 december 2016 om 10:00 uur

door

Huarong ZHENG

Master of Science in Intelligent Transportation Engineering,
Wuhan University of Technology, Wuhan, P.R. China
geboren te Jingshan, P.R. China.

This dissertation has been approved by the promotors:
Promotor: Prof. dr. ir. G. Lodewijks
Copromotor: Dr. R.R. Negenborn

Composition of the doctoral committee:

Rector Magnificus	chairman
Prof. dr. ir. G. Lodewijks	Delft University of Technology, promotor
Dr. R.R. Negenborn	Delft University of Technology, copromotor

Independent members:

Prof. dr. ir. E. Theunissen	Netherlands Defence Academy
Prof. dr. R. Dekker	Erasmus University Rotterdam
Prof. dr. ir. B. De Schutter	Delft University of Technology
Dr. C. Ocampo Martinez	Technical University of Catalonia
Prof. dr. ir. L.A. Tavasszy	Delft University of Technology

The research described in this dissertation is fully supported by China Scholarship Council under Grant 201206950021, and partially by the VENI project "Intelligent multi-agent control for flexible coordination of transport hubs" (project 11210) of the Dutch Technology Foundation STW, a subdivision of the Netherlands Organization for Scientific Research (NWO).

TRAIL Thesis Series T2016/25, the Netherlands TRAIL Research School
P.O. Box 5017
2600 GA Delft, The Netherlands
Email:info@rstrail.nl

Published and distributed by: Huarong Zheng
E-mail: zhr_19881211@126.com

ISBN 978-90-5584-218-6

Keywords: Waterborne AGVs, model predictive control, cooperative distributed control, robust control, closed-loop scheduling and control, Inter Terminal Transport.

Printed in the Netherlands

Preface

It has been almost four years since I landed at Schiphol Airport for the first time! I was told by the customs officer that the PhD study in the Netherlands would be boring when he asked about the reason why I came here. I was scared. However, there was no way back! Therefore, I headed on and started this uncertain journey. All kinds of incredible things, expected and unexpected, happened along the journey. Now, standing at the very end of my PhD and looking back, I can clearly see how different I am, which gives me the greatest sense of achievement, even greater than that from the research work presented in this thesis.

Four years ago, I did not know much about control. I spent quite some time following a master course, Control Theory, from the control department, but still failed the first exam. It was the first time in my study career to retake an exam. Nevertheless, I understood the field of control better and was able to develop some simple control algorithms after all. In small steps, I had my first simulation run successfully, and later even had some interesting results to present at internal meetings, international conferences, and even the world control congress. Acceptances for publication in high impact factor peer-review journals and the award in a poster session further confirm the research and industrial contributions of my work. Eventually, I finished this PhD thesis. At this moment, I owe a lot of thanks to those who have helped me make all these possible.

First of all, I would like to acknowledge the financial support from China Scholarship Council for my four-year living in the Netherlands, and from the Department of Maritime Transport & Technology, Faculty of Mechanical, Maritime and Materials, Delft University of Technology for all the research relevant expenses.

Secondly, I would like to thank sincerely my PhD supervisors, Prof. Gabriël Lodewijks and Dr. Rudy R. Negenborn. Prof. Lodewijks has always been critical about my research, which always makes me feel nervous before the meeting with him. Although he does not have many comments, those raised by him are always sharp and can pinpoint the deficiency of my algorithms. I benefit a lot from his rich academic and industrial experiences and insights into engineering problems. The group lunch times when he shared his life experiences all over the world are also the most enjoyable.

Dr. Negenborn is the one that I could not thank too much. As my daily supervisor, he has always been very patient, encouraging, and helpful in training me into an independent researcher. Every time we had a progress meeting, he would ask tens of questions, at the same time guiding me to the right research directions. Every time I sent him a draft paper, he would read and correct it so carefully. I am always the one to feel proud when a group of PhDs are comparing supervisors in terms of the quality and the waiting time of feedback on papers. His very organized working style has also impressed me deeply. I gained and learned so much by working with him, not only the knowledge necessary for my research,

but also the way of effective working and supervision. Definitely, I will benefit from these influences in my future academic career. Special thanks go to him for translating the thesis summary into Dutch.

Moreover, it has been a delight for me to work with all the colleagues in the Section of Transport Engineering and Logistics. I enjoyed all the lunch breaks together with them in Aula. I will not miss the food in Aula, but will, for sure, miss all the happy times we spent there.

In addition, I would like to thank all my Chinese friends in Delft who have encouraged and supported me all the way. Especially, I would like to mention Xiaoyan Wei and Xiangrong Wang, two lovely Chinese girls from whom I gained so much fun and positive energy. I will miss those times that we hang out together, shopping, movie or discovering nice Chinese food in Rotterdam and the Hague. It is their encouragements to exercise more that make me stay healthy physically during the PhD. My three housemates, Jinhu Wang, Changgong Zhang, and Dong Liu, who have excluded me from the list to take out the garbage, are the cutest housemates one could have.

Last but not the least, I owe my parents numerous thanks for their unconditional love, and my cute nephew for the laughter over the weekend video calls. I should also thank my boyfriend who has always been patient and tolerant when I poured my junk emotions to him.

I appreciate and treasure this precious experience in my life. It is the end. It is the start. Facing the future, I feel uncertain about everything again. However, I will head on again with more confidence this time.

Huarong Zheng,
Delft, August 2016.

Contents

Chapter 1

Introduction

Efficient and sustainable container handling is critical for large ports to improve competitiveness in the increasingly globalized economy. Terminal level operations in the port of Rotterdam have benefited significantly from innovative technologies such as Automated Guided Vehicles (AGVs) [135] and fully automated terminals. This dissertation proposes a new type of container transporter, the waterborne Autonomous Guided Vessels (waterborne AGVs) for smarter *port level* logistics. The main focus of this thesis is on developing control and scheduling strategies for coordinated waterborne AGVs to move containers autonomously between terminals, the so-called Inter Terminal Transport (ITT) [129].

In this chapter, the background and the motivation for the research on coordination of waterborne AGVs are first presented in Section 1.1. Section 1.2 formally defines the research scope and assumptions, followed by the research questions that will be addressed in this thesis in Section 1.3. This chapter is concluded in Section 1.4 with an overview of the contents of this thesis.

1.1 Waterborne AGVs for Inter Terminal Transport

In 2013, the Port of Rotterdam opened a new port area, Maasvlakte II, with approximately 2,000 hectares reclaimed from the sea [102]. Maasvlakte II together with Maasvlakte I forms a global container hub complex, as shown in Figure 1.1. It is the vision of the Port Authority to affirm its leading position in the field of efficiency and sustainability by 2030 [101]. Especially for the new port area, the Port Authority leaves space for innovative technologies aiming at developing it into the smartest and most sustainable port in the world. The port of Rotterdam has a long tradition of applying advanced technologies. The port of Rotterdam was the first port in the world adopting AGVs at the ECT Delta terminal in 1993. The first fully automated terminals, APMT, opened in 2015. A 40% increase in productivity is foreseen by APMT due to *automation* [3].

The port area of the Maasvlakte in Figure 1.1 has several features that are representative for general large ports. Firstly, it is expected that the throughput of containers will increase to more than 30 million Twenty-foot Equivalent Unit (TEU) per year by 2035 [101]. They need to be handled more efficiently in order to increase the competitiveness of the port. Movement of containers happens both inside terminals, likely handled by land-side AGVs,

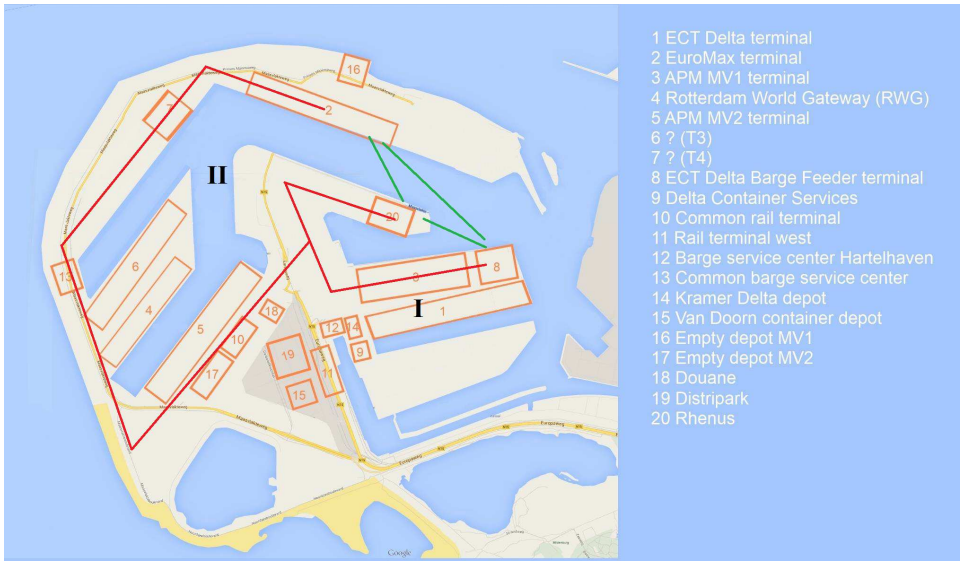


Figure 1.1: Overview of terminals at Maaasvlakte I and II; adapted from [32].

and between terminals via various modalities (e.g., road, rail, sea), i.e., ITT. At present, ITT is realized mainly by road vehicles, e.g., multi-trailer systems. Secondly, the reclaimed land from the sea in the new port area is limited, which poses challenges on expanding the existing physical transportation infrastructure to accommodate increasing traffic flow by land. Thirdly, for complex geographical layouts like the Maasvlakte, the distances between some terminals are much longer by land than by water. This is illustrated in Figure 1.1 by the example involving Terminals 8 (ECT Delta Barge Feeder terminal), 20 (Rhenus), and 2 (Euromax terminal). Distances among the three terminals by water indicated by the green lines are much shorter than by land indicated by the red lines. Last but not the least, the Maasvlakte can be seen approximately as a confined water area which has relatively unsophisticated traffic so far and has reliable advanced ICT systems supporting the development of intelligent infrastructures.

A new type of container transporter over water, waterborne AGVs, is proposed for ITT in this thesis. Literally related with conventional AGVs, waterborne AGVs share similarities but are different from conventional AGVs. Both waterborne AGVs and AGVs are unmanned software controlled vehicle systems for logistics distributions; and both of them are favorable for relatively simple environments with repeating transportation patterns. However, waterborne AGVs differ with AGVs on several aspects: *a*) waterborne AGVs are for transportation over water and AGVs for transportation over land; *b*) waterborne AGVs cannot be navigated by following markers, wires, or magnets etc. in the floor, which is the case for most of the existing AGVs. According to [87] on the definitions of levels of autonomy, waterborne AGVs have a higher level of autonomy than AGVs in that waterborne AGVs not only have built-in functionality but also have goal-directed reaction and behavior. Therefore, the word “Autonomous” has been used for waterborne AGVs while “Automated” for conventional AGVs; *c*) it is not possible to specify an area especially for waterborne AGVs

without external traffic as has been done for AGVs that are applied in manufacturing industries, warehouses, and container terminals [135]; *d*) temporal requirements for waterborne AGVs are more stringent than AGVs since for ITT, the most important criterion is "non-performance" which happens when the completion time of ITT tasks is later than the permitted latest arrival time [21].

In general, the potential benefits of developing waterborne AGVs for transport in port areas are summarized as follows:

- Waterborne AGVs could be almost labor cost free since no mariners are necessarily on board;
- Waterborne AGVs could offer another transport mode to handle the expected large throughput instead of exploiting the limited land in port areas for road traffic;
- Waterborne AGVs, comparable to land-side AGVs, could be optimally operated 24/7 with reliable performance and improve port efficiency exploiting automation;
- For terminals with longer distances by land than by water, waterborne AGVs could save energy compared to road vehicles; and
- Waterborne AGVs are in line with the development of smart ports and are deemed as very relevant to the ITT practice in the port of Rotterdam [24].

To develop a transport system using waterborne AGVs involves multi-discipline technological and methodological advancement. This dissertation focuses on the coordination of waterborne AGVs for ITT as to be further clarified next.

1.2 Scope and assumptions

We consider an autonomous waterborne ITT system: a fleet of waterborne AGVs that handles a set of ITT requests to transport autonomously specified amounts of containers between specified origins and destinations with temporal constraints in an energy efficient way.

Typically, a transportation decision-making system is hierarchically partitioned into three levels: long-term strategic, mid-term tactical, and short-term operational control decisions. Analogous levels of an autonomous ITT system using waterborne AGVs are shown in Figure 1.2. Within such an ITT system, a port authority runs a fleet of waterborne AGVs shuttling between terminals internally in the port area to transport containers. Strategic decisions regarding locations of berths for waterborne AGVs, fleet size, and composition issues, etc. are long term decisions in the order of years. Tactical and operational levels determine, for each waterborne AGV, the chronological events that occur at the hours time scale and the speeds, accelerations, or amount of power to input that occurs at the seconds time scale in order to assure those events are executed as scheduled, respectively. The tactical scheduling and operational control level problems as circled by the red dashed line in Figure 1.2 are of interest in this dissertation for coordinating waterborne AGVs.

Assumptions are made that:

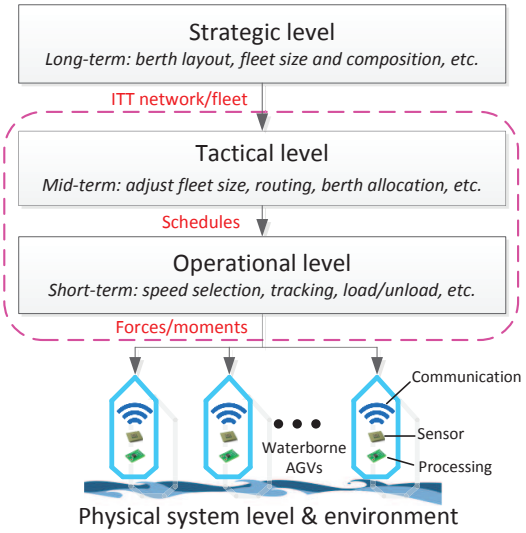


Figure 1.2: Different levels of an ITT system using waterborne AGVs; adapted from [16].

- The waterborne AGV fleet size and composition have been decided by the strategic level in a way that there is a sufficient number of waterborne AGVs available for ITT requests;
- The ITT network has also been designed at the strategic level. The network includes: berths that can accommodate waterborne AGVs by providing charging, maintenance, parking, etc., and routes as shortest paths connecting berths;
- Each terminal has one waterborne AGV berth with available load/unload equipment so that the berth allocation problem at the tactical level and the load/unload problem at the operational level vanish;
- Each waterborne AGV is equipped with sensors, communication devices, and processing units to measure its own system states, communicate with other waterborne AGVs within a certain range, and perform certain computations; and
- The number of containers that need to be transported from each ITT request is smaller than the capacity of waterborne AGVs, and split of ITT requests is not allowed.

From a control perspective, waterborne AGVs should be able to comprehend the surroundings and determine what to do autonomously in order to fulfill ITT tasks, i.e., pick-up and deliver containers at specified terminals at specified times. Challenges for the control level problems arise from various aspects: *a*) waterborne AGVs, like other marine surface vehicles, have limited maneuverability. Therefore, they cannot respond timely to environmental changes, which could lead to undesirable or even dangerous behavior, e.g., collisions; *b*) multiple conflicting operational control objectives exist including tracking, energy efficiency, low "non-performance" rate etc. even for one waterborne AGV; *c*) system constraints on inputs and outputs due to limited engine power, mechanical maximum

deflections/revolutions or spatial no-sailing zones, etc., need to be satisfied; *d*) complex waterborne AGV dynamics that model waterborne AGV behavior could render applicable control techniques limited; *e*) when multiple ITT tasks are scheduled for multiple waterborne AGVs, waterborne AGVs should preferably compute in a distributed and cooperative way; and *f*) if the influences of environmental disturbances, e.g., wind, waves, and currents, cannot be known perfectly beforehand, waterborne AGVs should react in a robust way and maintain safety, e.g., satisfying physical limitation and collision avoidance constraints, with possible reasonable performance deterioration.

From a scheduling perspective, the goal is to determine, for each waterborne AGV, a sequence of terminals to visit, the corresponding arrival times and loading/unloading volumes. The main challenge lies in developing a closed-loop and tightly integrated scheduling and control scheme for waterborne AGVs since the scheduling and control levels share the common aim of making economical and environmentally friendly decisions. Although both levels largely rely on mathematical models and optimization techniques and both aim at either maximizing profit or minimizing cost, the inherently different time-scale nature prohibits an integrated and computationally tractable solution.

1.3 Research questions

Following the scope and assumptions for the considered research problems with challenges presented above, this dissertation aims to *develop advanced control and scheduling strategies for coordinated waterborne AGVs applied to ITT*. This main research goal will be achieved by addressing the following five Key Research Questions:

1. Which technique is suitable for the control of waterborne AGVs?
2. What performance criteria should be considered in optimizing the process of one waterborne AGV carrying out one ITT task and how can the optimal performance be achieved?
3. How can multiple waterborne AGVs be coordinated for multiple ITT tasks with waterborne AGVs making decisions locally while minimizing the overall cost in a cooperative and distributed way?
4. How can environmental disturbances due to wind, waves, and current be systematically handled by cooperative and distributed waterborne AGVs?
5. In what way can the scheduling and control loop for waterborne AGVs be closed in order to obtain an energy-efficient autonomous ITT system?

For answering these Key Research Questions, a comprehensive literature review will be carried out, which also further motivates the research discussed in this thesis. System and control theories, model predictive control (MPC) in particular, will be utilized extensively. Waterborne AGV dynamic trajectories considering various system constraints will be controlled and optimized satisfying possibly conflicting design objectives using deterministic, distributed, and robust control tools. Coordination at the tactical level will also be considered and will be tightly integrated with the control problems to build an autonomous ITT system using waterborne AGVs.

1.4 Thesis outline

The road map of this dissertation is presented in Figure 1.3, illustrating connections of chapters and a suggested order in which the chapters can be read. The contents for Chapters 2 – 8 are summarized briefly as follows:

- **Chapter 2** reviews relevant literature on control and scheduling techniques that are applied to improve intelligence of ground vehicles and marine surface vehicles. Existing approaches and applications regarding MPC, distributed MPC, and robust MPC techniques are discussed. This chapter will partially answer Key Research Question 1.
- **Chapter 3** defines the dynamic models for waterborne AGVs that are used in the latter chapters of this dissertation. A nonlinear dynamic model of marine surface vehicles with three degrees of freedom is used to simulate waterborne AGV behaviors. Successively linearized dynamic models based on this nonlinear model are used to predict waterborne AGV trajectories over a future horizon for controller development. This chapter will partially answer Key Research Questions 2 – 5.
- **Chapter 4** proposes a predictive path following with arrival time awareness controller for one energy-efficient waterborne AGV. Control goals will be achieved by a proposed connected reference path coordinate system, a switching logic for avoiding overshoots, and a two-level double integrator scheme being aware of the arrival time. This chapter will partially answer Key Research Question 2.
- **Chapter 5** extends the proposed controller for one waterborne AGV in Chapter 4 to scenarios involving multiple waterborne AGVs. Cooperative distributed waterborne AGVs will be realized using the alternating direction method of multipliers (ADMM). A fast ADMM algorithm is further proposed to improve convergence rates. This chapter will partially answer Key Research Question 3.
- **Chapter 6** considers the influences of environmental disturbances and proposes a novel cost-effective robust and cooperative distributed control approach for multiple waterborne AGVs. We define system robustness levels, propose and solve a cost-effective robust distributed MPC problem for waterborne AGVs based on results from Chapter 5. This chapter will partially answer Key Research Question 4.
- **Chapter 7** closes the loop of scheduling and control of waterborne AGVs for an autonomous energy-efficient ITT system. A novel pick-up and delivery scheduling approach considering safe intervals between berthing time slots of different waterborne AGVs is also proposed. This chapter will partially answer Key Research Question 5.
- **Chapter 8** concludes the research in this thesis and outlines directions for future research.

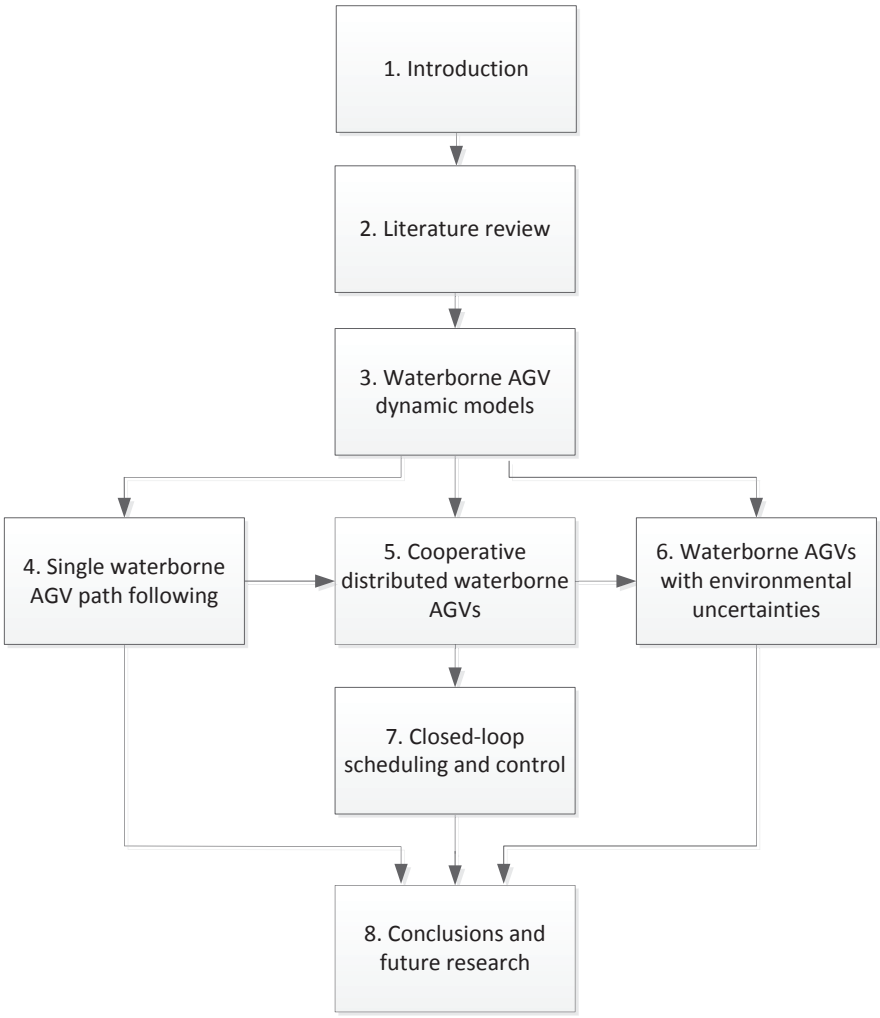


Figure 1.3: Road map. Arrows indicate read before *relations.*

Chapter 2

Literature review and background

This chapter presents an overview of the literature relevant for the development of waterborne Autonomous Guided Vessels (waterborne AGVs). Section 2.1 introduces two types of intelligent vehicles, land-based Automated Guided Vehicles (AGVs) and intelligent marine surface vehicles, which are closely related to waterborne AGVs. Coordinating technologies regarding motion control and scheduling that could be applicable to waterborne AGVs are then subsequently reviewed in Section 2.2 and Section 2.3, respectively. Section 2.4 discusses briefly the research work on Inter Terminal Transport (ITT). Conclusions of this chapter are presented in Section 2.5.

2.1 Relevant intelligent vehicles

Broadly speaking, intelligent vehicles should possess certain capabilities in the so-called observe, orient, decide and act loop [10]. Waterborne AGVs belong to the class of intelligent vehicles since they are able to observe the surroundings and determine what to do autonomously to fulfill ITT tasks. This section introduces the development of two types of existing intelligent vehicles, land-based AGVs and intelligent marine surface vehicles, which are closely related to waterborne AGVs.

2.1.1 Automated guided vehicles

Land-based AGVs are driverless vehicles that were first introduced in the 1950s simply towed by a chain [83]. They came into industrial and commercial use in 1976 by adopting invisible markers on the floor [23], and saw wide applications in the late 20th century [135]. They are especially designed to move materials in certain areas, e.g., manufacturing industries, warehouses, and container terminals with a certain level of intelligence. Specifications of AGVs differ from one another depending on their application scenarios. The proposed waterborne AGVs can be seen as an extension of AGVs in container terminals in the sense that AGVs move containers internally in terminals over land and waterborne

Figure 2.1: The ECT Delta Terminal Rotterdam (Courtesy of ECT).

Figure 2.2: Terex Gottwald AGVs [128].

AGVs move containers between terminals within a port area over water. Therefore, we confine our introduction to AGVs to those applied to container transportation in container terminals.

The first container carrying AGV was adopted at the ECT Delta terminal (see Figure 2.1), in the port of Rotterdam in 1993, almost 40 years later than its first introduction and almost 20 years later than its commercial use in other industries. Ever since then, AGVs have been widely used in semi- and fully automated container terminals. Cleaner, safer, and more efficient AGVs are developed to improve container terminal internal automation and efficiency. Figure 2.2 shows currently adopted AGVs working in container terminals.

In a transport system using AGVs, four parts are identified as fundamental [135]: *1*) the vehicles; *2*) the transportation network; *3*) the physical interfaces (pick-up/delivery points) that link the storage and transport systems; and *4*) the coordinating system. A number of essential decisions have to be made for each part when designing an AGV system. In terms of

vehicles, the problem investigated the most is the minimum fleet size [136] required for the system to, on the one hand, satisfy transport demands, and on the other hand, be economical and avoid congestion. Other vehicle design issues include deciding on features related to capacity, speed, power, costs, and more recent self-lifting or non-lifting [21], fixed path or free-ranging [143], which all interact with the fleet size problem and other system parts. The transportation network in container terminals connects pick-up (quay side) and delivery (stacking area) points [142] and defines the guide paths that AGVs follow. Markers or wires in the floor are usually necessary for navigating AGVs through the network. Decision problems in physical interfaces concern location and equipment choices [119]. Quay cranes and stacking cranes are commonly used for loading and unloading containers to and from AGVs, respectively.

The last part of an AGV system, the coordinating system, is critical in guaranteeing system performance, e.g., maximizing throughput, minimizing makespan, delays, and cost while avoiding conflicts among AGVs. Collisions and deadlocks [135] are the two likely conflicts for fixed path AGVs. A proper layout of the guide paths could prevent conflicts, which could degrade system performance though. Two more commonly employed approaches are the so-called zone-control [44] and advanced scheduling strategies [130]. The idea of zone-control is similar to using traffic lights at crossroads by dividing the network into several control zones and allowing one vehicle to occupy a zone at one time. Although simple to implement, zone-control could also sacrifice system performance. Extensive research has been done on scheduling problems that aims at constructing routes an AGV can take and schedules that give times when the AGV can traverse corresponding routes, see [135] and references therein. Operations research based vehicle routing problems (VRPs) [130] or flow shop problems [142] are widely used to formulate the scheduling problem either in a static or dynamic way. However, the existing coordination mostly takes place at the scheduling level (see Figure 1.2), and is only applicable to AGVs with fixed paths and perfect executions of schedules in a disturbance free environment; dynamics that model the movement of AGVs are rarely considered by the scheduling problems in the literature. In [142, 143], hierarchical approaches are proposed for scheduling and trajectory planning of free-ranging AGVs. One dimensional point-mass dynamics are used to model free-ranging AGVs.

2.1.2 Intelligent marine surface vehicles

Intelligent marine surface vehicles using automatic co-pilots or even autopilots require less human supervision or intervention during the voyage. Higher intelligence directly reduces the workload of mariners, and lower the rate of human errors which are the most important factor contributing to maritime accidents [42].

In general, intelligence is achieved by sensing the environment, processing the sensed data, and determining what to do based on given missions and current system states. Intelligent marine surface vehicles are usually equipped with a navigation, guidance, and control (NGC) software system that is responsible for those capabilities. Figure 2.3 briefly illustrates the technological components of such an NGC system. The navigation system deals with noised or even incorrect measured (partial) system states from various sensors and obtains estimated states for use in other components. Commonly used sensors for intelligent marine surface vehicles include positioning devices like Global positioning system (GPS),

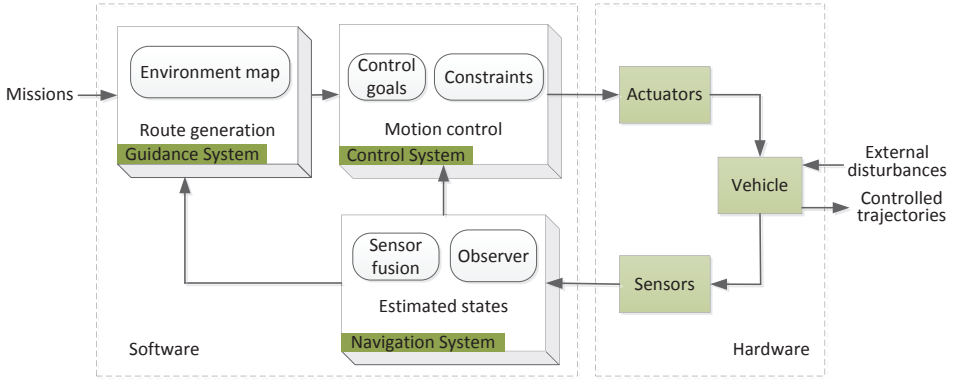


Figure 2.3: Diagram of an NGC system for intelligent vehicles (adapted from [30]).

more accurate Differential GPS, or higher cost inertial navigation systems, vision sensors using cameras, and maritime detection equipment such as wave or depth sensors [14] using radars. Radio, Wi-Fi, and Automatic Identification System (AIS) are also seen installed on maritime vehicles [147] for vehicle-to-vehicle or vehicle-to-control center communication. The guidance system generates reference routes based on environmental information and given missions. References are then tracked by the vehicle which is controlled by the control system. Mostly, there exist possibly conflicting control goals such as tracking accurately, arriving at the destination on time, and using an as small amount of energy as possible. The control system also takes care of system dynamical limitations such as maximum maneuvering speed, maximum engine deflections or revolutions. The processing in guidance and control systems relies on theoretical tools of optimization and automatic control as to be reviewed in the following sections.

Research work on intelligent marine surface vehicles has always been active ever since the first autopilot for ships was proposed in [79]. Platforms and prototypes are built for various purposes. Table 2.1 provides an overview of the developed prototypes by different organizations worldwide. NGC software associated with prototypes is also developed for intelligent marine vehicles. The *SCOUT* team developed a set of open source modules MOOS-IvP [55] for providing autonomy to general robotic platforms. Including the *Cybership II* system, shown in Figure 2.4, a comprehensive software library called Marine Systems Simulator [99] is developed to provide necessary resources for rapid implementation of mathematical models and controllers for marine systems. Vehicles like *SCOUT*, *Springer*, *DelfimX*, and *Delfia-1* (Figure 2.5) are designed to be operable in both remote control and fully autonomous control modes. However, high speed military vehicle *PROTECTOR* still relies significantly upon operator guidance and remote control. Most vehicles are equipped with one PC for all the processing tasks in the NGC system while *Springer* has three PCs on board, each responsible for a module, i.e., navigation, guidance, and control. Note that, except for *Delfia-1*, few of the aforementioned intelligent marine vehicle platforms have been designed in the context of transport and logistics.

Table 2.1: Prototypes of intelligent surface vehicles.

Organization	**Vehicle**	**Purpose**	**Year**
MIT Sea Grant Program, US	*ARTEMIS*, *ACES*, *AutoCat*, and *SCOUT* [17]	Oceanographic data collection, research	1993, 1997, 1999, 2004
Instituto Superrior Tècnico, Portugal	*Delfim*, *Caravelas*, *DelfimX* [34]	Coordination with underwater vehicles	2004
University of Rostock, Germany	*Measuring Dolphin* [72]	Shallow water survey, oceanography tasks	2006
University of Plymouth, UK	*Springer* [126]	Environmental and geographical survey	2007
Institute of Intelligent Systems for Automation, Italy	*Charlie* [13]	Mine hunting	2003
Eotvos Lorand University, Hungary	*Sillverlit* [127]	Research	2011
Israeli Rafael Advanced Defense Systems	*PROTECTOR* [108]	Military	2005
United States Navy	*Spartan Scout* [112]	Military	2001
Marine Cybernetics Laboratory, Norwegian University of Science and Technology	*Cybership II* [121]	Research	2005
Delft University of Technology, the Netherlands	*Delfia-1* [88]	Research	2015

Figure 2.4: Cybership II at Norwegian University of Science and Technology [121].

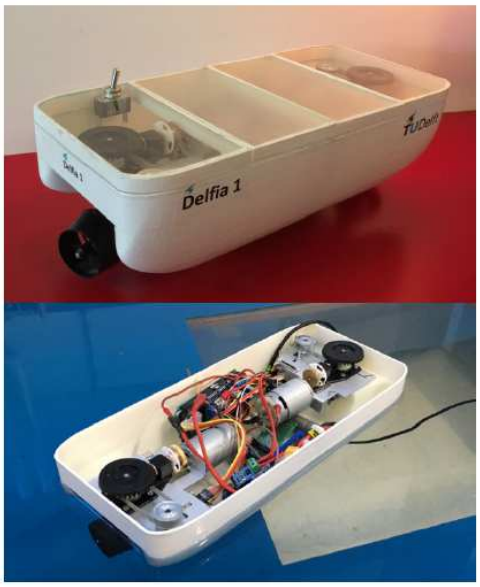


Figure 2.5: Delfia-1 at Delft University of Technology [88].

2.1.3 Summary

Several conclusions can be drawn from the development of land-based AGVs and intelligent marine surface vehicles for waterborne AGVs. Firstly, the design of a transport system using waterborne AGVs are similar with a system using AGVs. Decision problems on the four system parts, i.e., the vehicles, the transportation network, the physical interfaces, and the coordinating system could be comparably identified. Secondly, since all the problems are essentially coupled and interrelated, simultaneous decisions are preferable for a transport system. However, hierarchical approaches are generally proposed in the literature for tractable solutions. Tighter integration of the hierarchical levels could be expected to achieve economical benefits. Thirdly, few research on intelligent marine surface vehicles has been targeted for civilian use or transportation. Safer, more sustainable, and efficient marine transport systems could be expected if more intelligent waterborne transport vehicles are used.

2.2 Motion control literature

Waterborne AGVs are operated by controllers instead of human beings. When applied to ITT, it is desirable that the waterborne AGV controller can achieve the following goals:

- A given geometric reference path could be tracked with deviations as small as possible;
- A given arrival time requirement could be met when a preferable time is feasible considering system limitations, or a minimal delay with respect to the preferable time

within a specified time window otherwise;

- The aforementioned two design requirements could be achieved in an energy economical way;
- Distributed decision making could be possible even when there exist couplings among waterborne AGVs; and
- A certain level of robustness could be achieved when uncertainties are involved.

In this section, we review general control techniques that have been applied to marine surface vehicles, and that could be applicable to waterborne AGVs for controllers that meet the above specifications. Particularly, we review model predictive control (MPC) which will be used extensively in the later chapters.

2.2.1 General control techniques

The meaning of "control" refers to achieve desired system dynamical behaviors using designed algorithms in this thesis. The system together with the controller is called a controlled system. Since external disturbances, mismatches between the model and the system, and variations in the system itself ubiquitously exist, control engineers generally prefer feedback (closed-loop) control with inherent robustness over open-loop control [4]. Briefly, a feedback controlled system measures system outputs, compares the outputs with references, computes corrective inputs probably based on a system model, and applies the inputs to the system to achieve desired behaviors. However, design and analysis of feedback controllers can vary significantly depending on control purposes and system characteristics such as whether the system is deterministic or non-deterministic, with linear or nonlinear dynamics, and with coupled or decoupled subsystems.

Particularly, for motion control of marine surface vehicles, three categories of control problems with different purposes are recognized [30]:

- **Setpoint regulation**: In this case, the references to the controlled system are constant, and the corresponding controller is also called a regulator. Examples are constant speed regulation [29] , heading control [62], and dynamic positioning [124]. Regulation control is also one of the most widely analyzed and best understood problems in the field of automatic control.
- **Path following**: The reference is a geometric path independent of time. The reference path can be straight lines [28, 93, 150] or curves [19, 139] without temporal constraints.
- **Trajectory tracking**: Explicit time parameterized references (e.g., positions, velocities) must be given. The control goal is then to drive the system to the specified states at specified time [146].

Different types of control techniques are applied to the above motion control problems for marine surface vehicles:

- The first recognized and most widely implemented controller until now is proportional-integral-derivative (PID) [79] which was first proposed for ship steering control. PID has the advantages of being simple to implement and at low cost [134]. However, issues such as parameter tuning, overshoots, constraints, and performance guarantees are recognized in PID design.
- Lyapunov-based control design and analysis are more systematic and sophisticated. Analytical control laws are usually available with guaranteed Lyapunov stability [4] for controlled marine surface vehicles [19, 28, 122]. In [19, 28, 122], the analysis is done based on derived error dynamics. Specifically, the reference path given in [19] is second-order time differentiable so that second-order system dynamics can be converted to error dynamics. In [28], the references are straight line segments. A Line-of-Sight (LOS) guidance method is introduced to provide moving references for the system to track. A maneuvering problem is proposed in [122] where a geometric task and a dynamic task are involved. The geometric task guarantees path convergence and the dynamic task tracks an assigned speed along the path. Constraints and performance regarding cost are not considered in these approaches.
- Sliding mode control, as a nonlinear control technique, has also seen applications [9, 38] to marine surface vehicles which have complex nonlinear dynamics. Sliding mode control laws are discontinuous and can have "chattering" phenomenon. Moreover, constraints and cost performance cannot be systematically considered as well.
- Intelligent control methods, e.g., fuzzy logic [91], neural network [15] and genetic algorithm [84] are model free and are based on heuristics. Usually an intelligent and a non-intelligent control methods are combined. In [84], the genetic algorithm is used as an optimization tool in an MPC framework. Challenges with intelligent control are that heuristics are generally empirically determined, e.g., the fuzzification and defuzzication rules for fuzzy logic control [91], and system properties are difficult to be analyzed.
- Optimal control [31] differs from other control techniques in that it can formulate a particular *objective function*, and thus achieves desired behaviors at an optimal cost. The intelligent marine surface vehicle prototype *Springer* is controlled by a linear-quadratic-Gaussian controller [85]. Simplified linear vehicle dynamics are used. Broadly speaking, MPC also belongs to optimal control. System constraints and design indices are explicitly taken into account for marine surface vehicle motion control problems using MPC in [62, 139, 150]. However, optimal control relies on solving mathematical optimization problems which can be hard when nonlinear system dynamics and constraints are present.

An overview of the characteristics of the discussed control techniques are summarized in Table 2.2.

2.2.2 Model predictive control

MPC is one of the most widely implemented control techniques in process industries [107]. General advantages of MPC include handling system constraints systematically and optimizing system performance quantitatively. In particular, considering the characteristics and

Table 2.2: An overview of different control techniques applied to marine surface vehicles.

Control technique	**Advantage**	**Disadvantage**
PID	Simplicity and computationally fast	Parameter tuning, overshoots; constraints and performance not guaranteed; non-predictive
Lyapunov-based	Analytical control laws and theoretical properties	Assumptions on system characteristics; constraints and performance not guaranteed; non-predictive
Sliding mode	Analytical control laws and robust to disturbances	Chattering phenomena; constraints and performance not guaranteed; non-predictive
Intelligent methods, e.g., fuzzy, neural, genetic algorithm	Model free, simple and computationally fast	Relying on experiences; performance not quantified; non-predictive
Optimal control	Constraints and performance guaranteed; easy tuning	Model-based; computational issues with complex models

the main challenges in motion control problems of waterborne AGVs as discussed in Chapter 1, the following justifications for the suitability of applying MPC to waterborne AGVs are made:

- Waterborne AGVs are with limited maneuverability and could not respond to environmental changes or emergencies timely. MPC makes decisions based on predicted information over a future horizon. Therefore, it can anticipate and prevent undesirable and dangerous situations, e.g., likely collisions, at an early stage;
- Waterborne AGVs have physical limitations on input, states, and outputs, e.g., maximum speed and engine power. Moreover, waterborne AGVs need to be a safe distance away from each other. MPC can handle these constraints explicitly and systematically;
- Waterborne AGV systems have multiple inputs and outputs, which can also be handled elegantly by MPC;
- For small magnitude of modeling inaccuracies and external disturbances, MPC has inherent robustness;
- The desired behaviors of waterborne AGVs are defined considering possibly conflicting safety, economical, and environmental factors. Optimization based MPC makes optimal and quantified trade-off among conflicting objectives with respect to user defined criteria.

However, there are also concerns with MPC applications. Firstly, although MPC solves a finite horizon optimization problem, which relieves computational burden in infinite horizon optimal control [31], optimizations in MPC mostly are necessarily solved online repetitively except for explicit MPC [8] which is, however, only applicable to simple low order

Algorithm 2.1 Basic MPC algorithm

1: Measure system states at step k;
2: Predict system trajectories based on the system model;
3: Formulate and solve the optimization problem;
4: Apply the first control input;
5: $k \leftarrow k+1$ and go to Line 1.

dynamics. This restricts the online optimizations to those that can be solved efficiently. Secondly, controlled system properties, e.g., recursive feasibility, stability and robustness, are extensively analyzed and guaranteed by design only for certain classes of systems, in particular for linear time-invariant systems. For more complex systems not necessarily respecting assumptions made in theoretical analysis, there are still open issues regarding theoretical properties.

Next, considering the motion control problems for a singe waterborne AGV, multiple cooperative waterborne AGVs, and waterborne AGVs with environmental disturbances, we review relevant research on MPC for individual systems, networked systems, and systems with uncertainties in the literature.

Individual systems

For individual deterministic systems, MPC solves online optimization problems obtaining a sequence of optimal control inputs and applies the first control input to the system in a receding horizon way. The predicted system trajectories over a prediction horizon are driven by solving the optimization problem to the desired behavior, as shown in Figure 2.6. Five essential components are recognized for an MPC controller design [110]: *1*) System prediction model; *2*) Cost indices; *3*) Constraints; *4*) Solving optimization problems; and *5*) Receding horizon principle. Then a general MPC controller can be designed using these five components as Algorithm 2.1.

Nominal stability and recursive feasibility properties of Algorithm 2.1 have been well known since 2000 for specific cases when the system model, cost function, and constraints satisfy certain conditions, see [36, 76, 77] and references therein. Generally, regulation or stabilizing problems are considered. Properly defined cost functions and constraints are necessarily present so that the "energy" of the system can be proved to dynamically decrease. System states are then guaranteed to converge to desired states and the feasibility at a previous step immediately implies the next step feasibility [110]. Linear time-invariant systems are extensively studied [77]. Properties of MPC for nonlinear systems in certain conditions have also been well understood [36]. Various extensions of the basic MPC in Algorithm 2.1 exist including hybrid MPC [58], economic MPC [109], explicit MPC [8], distributed MPC [71] and robust MPC [78]. Applications of MPC in practice, however, usually do not satisfy the cost format and assumptions made for theoretical analysis. Moreover, designed extra constraints that guarantee theoretical properties could probably degrade system performance. Successful applications of MPC are widely seen in process industries [107], intelligent cars [138, 145], power networks [90], intelligent transport systems [60], automated container terminals [143], and intelligent buildings [68] mostly without "stablizing" ingredients.

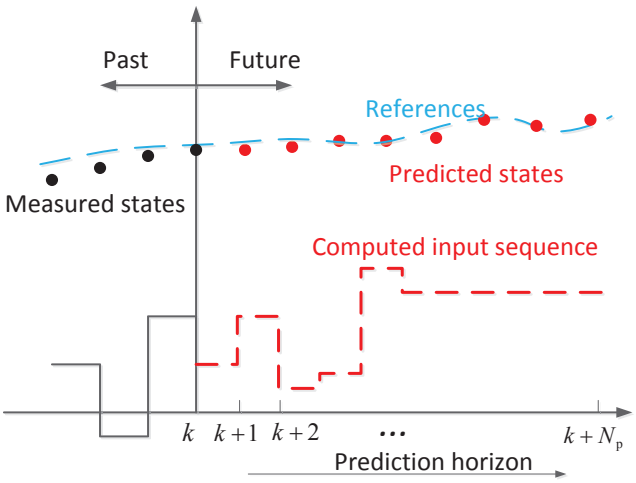


Figure 2.6: Illustration of MPC.

The first application of MPC to marine vehicles in literature is [137]. An MPC controller based on nonlinear vessel dynamics is designed to track splines representing waterways. Computer simulations and experiments on inland vessels show the effectiveness of the MPC controller. Recently, MPC has been applied to vessel path following [63] and heading control [62]. In [93], the LOS guidance [28] is integrated in MPC which uses a linearized model. When the vessel's heading angle is controlled to converge to the angle provided from the guidance module, cross-track errors can be proved to converge to zero realizing path following. However, unrealistic assumptions that velocities and cross-track errors should be small are made. A similar vessel path following problem is considered in [98] combining MPC and an LOS guidance law. In both [98] and [93], reference paths are specified by waypoints. Overshoots during switching of waypoints are observed due to the use of non-predictive reference information. Few works consider hazardous area avoidance or timing issues.

Networked systems

In general, system-wide control for networked systems can be approached in four ways [89, 116]:

- centralized: there is a single controller taking care of the entire networked system;
- decentralized: there are multiple controllers and each controller solves a local subsystem control problem using only local information, not relying on communication between subsystems;
- distributed: there are multiple controllers and each controller solves a local subsystem control problem, using also communicated information from other subsystems;
- hierarchical: controllers are working at different levels possibly with different time scales.

For many applications, distributed approaches are preferable [71]. Particularly for multiple waterborne AGVs when coupled by collision avoidance constraints for safety, we observe six reasons for this preferability: *1*) physically distributed by nature; *2*) computational efficiency; *3*) limited communication range; *4*) modularity for maintenance or expansion of the system; *5*) privacy issues if with different ownerships; and *6*) robustness to local failures.

Solutions to distributed collision avoidance include priority [86], potential field [114], velocity obstacles [54] and optimization [53] based methods. Conflicts disappear if subsystems are prescribed relative priorities by certain rules (e.g., COLREGs) [86], but fixed rules degrade system flexibility and optimality. Potential field approaches model conflicts as repulsive forces [114] for which it is difficult to consider multiple objectives and constraints. Velocity obstacles [54] usually assume constant velocities which might not hold in complex situations. Optimization based approaches are largely embedded in distributed MPC [71] considering the advantages of MPC as listed in Section 2.2.2.

In the literature, many distributed MPC approaches have been proposed with applications to various networked systems, e.g., power grids [90], aerial vehicles [53], intermodal freight transport [60], and traffic networks [64]. Few applications of distributed MPC to marine surface vehicles are seen in the literature to date. An overview of 35 different distributed MPC approaches categorized by process, control architecture, and theoretical properties is provided in [71]. Much of the research realizes distributed control by solving local problems sequentially using communicated intent trajectories from coupled neighboring subsystems [22, 47, 50, 53, 59, 65, 111, 132, 133]. Assumptions are then made that the deviations of communicated intent trajectories from actual trajectories are either small [53, 111, 132, 133], or compatibility constraints [22], penalty functions [59], bounds [65] are introduced to *make* the deviations small. With knowledge of the overall system dynamics, each local controller solves a centralized problem in [50]. The coupling effects from other subsystems are treated as bounded disturbances in [47]. Subsystems coupled via inputs are treated as agents in a cooperative game and distributed control is achieved following game theory in [70]. However, generally in sequential approaches, when one subsystem is computing, other subsystems are idle; and the order or priorities of computing agents still matters.

In terms of control architecture, besides computing in sequential, distributed MPC can also be achieved in parallel [90]. Parallel distributed MPC treats all agents equally and usually iterative negotiations between subsystems are required before an overall agreement is reached [26, 90, 95, 131, 148, 149]. For systems with special structures, the separability of dual decomposition is exploited to realize parallelism in [33] with accelerated convergence rate. For more general system couplings, [26] provides two distributed MPC solutions, i.e., dual decomposition and the alternating direction method of multipliers (ADMM) [12] which has better convergence properties than dual decomposition. Conventionally, ADMM is formulated as a 2-block consensus problem of which proof of convergence has been well established [12]. The 2-block consensus ADMM has been applied to the flocking problem achieving near-centralized performance [125], communication networks to reduce congestion [82], and networked road vehicles achieving the inner loop optimality in a two-loop convex-concave procedure [95]. A large number of iterations are usually required before convergence to a modest accuracy is obtained. Variants of ADMM, e.g., multi-block schemes [81] and varying penalty parameters [39], show convergence in numerical simulations for specific applications though with less rigorous convergence theorems.

Systems with uncertainties

System robustness against uncertainties with performance guarantees and constraint satisfaction are dealt with in the literature in several ways. Besides systematically considering system constraints and optimizing performance, MPC, under certain conditions, is inherently robust by solving repetitively online optimization problems with new system outputs [76]. However, this inherent robustness can only handle sufficiently small uncertainties [49]. A more reliable approach is to have designed robustness. For bounded uncertainties, an intuitive option is min-max MPC [47] minimizing a worst-case performance index; conservativeness and prohibitive computational time are the concerns. Alternatively, constraint tightening based MPC [53, 78, 111, 131] has a comparable complexity as that of conventional MPC by solving nominal optimization problems with tightened constraints. In [62] for ship heading control in wave fields, disturbances are estimated and compensated in a two-step MPC algorithm, assuming bounded estimation errors.

In practice, however, disturbances such as environmental forces acting on marine vessels often bear stochastic characteristics [30], are not necessarily bounded, and even when bounded, the bound is typically unknown. For unbounded uncertainties, small constraint violations are necessary either via soft constraints [144] or stochastic MPC [52, 104]. Two formulations of stochastic MPC are the expected case [104] and the chance constrained case with a specified probability of constraint satisfaction [52]. However, similarly with bounded uncertainties, this probability is still pre-designed. The trade-off between specified uncertainty probabilities and system performance is investigated in an Antarctic krill catch level control problem [43], showing in results from multiple simulations that increasing constraint satisfaction probability leads to an exponential decrease of catch levels. Bounds of a subset of uncertainties with high confidence are first determined with a scenario approach and then utilized in a robust problem in [74]. The bounds are, however, still fixed in optimizations.

Regarding robustness in distributed MPC, efforts have been made on decomposing the overall uncertain system based on distributed approaches reviewed in Section 2.2.2 and solving local robust MPC problems based on robust approaches reviewed before. Bounded coupling effects from other subsystems are considered in min-max local robust problems in [47]. By assuming small deviations of communicated intent trajectories from actual trajectories, [53, 111] solve local constraint tightened nominal problems serially. The intent and actual trajectory deviations are explicitly penalized in distributed cost functions in [59] and are explicitly bounded in [65]. External disturbances are then accommodated together with the coupling penalty function and coupling bounds by robustness constraints in local problems in [59] and [65], respectively. For linear systems with coupled state constraints, different tube-based robust distributed MPC (RDMPC) problems have also been proposed, e.g., single-update scheme [132], parallel-update scheme [131], and hierarchical control schemes [113]. Most aforementioned approaches solve local robust problems serially except for [131] which, however, loses cooperativeness. Note that iterative parallel approaches [33, 148] have only been studied in deterministic cases.

2.2.3 Summary

There are several options of control techniques that could be applicable to motion control problems in general. MPC turns out to be the most suitable for controlling waterborne

AGVs with special purposes in the context of transport and logistics. Extensive research on both theoretical analysis and applications of MPC, distributed MPC, robust MPC, and RDMPC has been done. More advanced techniques still need to be developed for the particular cases of a single waterborne AGV, networked waterborne AGVs with cooperative distributed solutions, and waterborne AGVs with environmental disturbances.

2.3 Scheduling approaches

Scheduling in this thesis, as shown in Figure 1.2, refers to determining the reference information for the lower level control problem. The reference information contains, for each deployed vehicle, a sequence of routes to travel and chronological events that occur along the routes. Approaches that are relevant for scheduling waterborne AGVs are reviewed in this section.

Essentially, waterborne AGV scheduling for ITT is a pick-up and delivery problem (PDP) [115] with time windows using capacitated vehicles. PDP is a generalization of a VRP [130]. Both PDPs and VRPs involve finding a set of optimal routes for a fleet of vehicles but differ in that PDP deals with transportation between distinct pick-up and delivery locations while in VRP, either the pick-up or the delivery location needs to be the same, i.e., the depot. Within the operational research realm in a logistical context, it is customary and sufficient to only care for setting schedules on discrete events. Details on how these events really happen, i.e., the evolution of the lower level system dynamics, are generally neglected. From a control point of view, however, vehicles concerned in VRPs or PDPs are actually assumed as dimensionless mass points predominantly with constant speeds such that any lower level feedback becomes irrelevant in a scheduling problem. We observe that two variants of VRPs are exceptional. The time-dependent VRPs [27] adopt a time-dependent speed model which, to some extent, considers lower level information, e.g., traffic congestion. But the speed is known a priori rather than being a decision variable that could be manipulated. The time-dependent VRPs belong to a more generic class of dynamic VRPs [106] dealing with dynamism such as online requests, dynamic travel times, etc. and update route responsively. Solutions with acceptable quality and computational efficiency are largely of concern for dynamic VRPs. Exact solutions are generally only applicable to small networks [120]. Therefore, considerable research has been done on developing heuristic methods to solve large network dynamic VRPs efficiently [57]. The second exception is the pollution-routing problem proposed in [7], which considers factors as load and speed in producing "environmental-friendly" vehicle routes. The resulting problem is more difficult to solve but yields lower load and speed dependent energy consumption cost. Still, the combined route-speed optimization is open-loop and far from being able to consider lower level complex dynamics.

In the maritime sector, the relation between marine vehicle speed and energy consumption is highlighted even more by both practitioners and researchers. The engine of Maersk "Triple-E" [69] was designed to sail relatively slowly to reduce 50% of the CO_2 emitted on the Asia and North Europe transport route. Another common practice in the shipping industry known as "slow-steaming" [73] by cruising at a lower speed than the design speed to reduce cost has also been widely accepted and implemented [105]. Arrival times are optimized in [25, 92] to obtain optimal speeds along shipping routes. Results of applying

the method to real shipping routes shows the potential for reducing environmental emissions is substantial. Besides the emphasis on speed, coordination of arrival times of ships at terminals to avoid unnecessary waiting or conflicts is more critical than for land-based vehicles. The reasons are twofold. First, ships visit the same terminal more frequently considering the limited pick-up and delivery locations. This is particularly the case in ITT. In fact, most PDPs assume distinct pick-up and delivery locations and each vehicle visits each location exactly once [115], which diminishes the arrival time coordination. Secondly, loading/unloading of ships could take more time than land-based vehicles, and thus cannot be neglected. Berthing time clash avoidance is modeled in [96] by constraining, for pick-up and delivery visits sharing a same berth, the departure time of a visit not to be larger than the arrival times of a later visit. This is problematic when extra time intervals are imposed between departure and arrival times which is practically the case if ship dimensions and safety distances are considered. Another characteristic of maritime logistics is that environmental uncertainties are prevalent. These uncertainties include current, waves, wind and encounters with other moving objects that not only interact with waterborne AGV dynamics at the operational level but also influence the scheduling level. This calls for a closed-loop system that makes decisions based on real-time feedback with tightly integrated scheduling and control levels.

However, scheduling and control, typically as two distinct levels in a transportation decision-making hierarchy, have been explored independently by researchers in the two areas [61]. Although both levels largely rely on mathematical models and optimization techniques and both aim at either maximizing profit or minimizing cost, the inherently different time-scale nature brings technical challenges for an integrated and computationally tractable solution. On the one hand, discrete decisions involved in scheduling problems restrict them to nothing but low dimension models solved in low frequency and off-line; on the other hand, feedback and closed-loop operation in real-time are essential in control systems to handle disturbances and complex dynamics. Efforts have been made either from a "Top-down" perspective by considering control elements in a scheduling problem [142] or from a "Bottom-up" perspective by including scheduling-oriented economic terms in the cost function of a control problem [2]. In the field of process industry, the economic benefits of integrating scheduling and control have been recently recognized and emphasized [5]. A so-called "time scale-bridging" model is proposed in [20], but this model counts on an explicit, low-order representation of the input/output process dynamics which is by all means hard to derive for general systems. Moreover, operational constraints cannot be incorporated. A decent solution to integrated scheduling and control has to date not yet been proposed.

Summarizing, the scheduling problem of waterborne AGVs can be formulated as a PDP. Special issues for waterborne AGVs in port areas include emissions and coordinated berthing times. A closed-loop scheme with more tightly integrated scheduling and control could be expected to benefit the overall coordination performance of waterborne AGVs.

2.4 Inter terminal transport

Waterborne AGVs are especially proposed and designed for ITT to improve the *port level* autonomy and efficiency in logistics. We give a brief introduction to ITT in this section.

ITT refers to the transportation of goods between terminals including dedicated auxiliary and value-added logistics service areas (e.g., inventory, packing, cargo bundling, repairing, and cleaning) within a port [41]. ITT can be conducted either by land using trucks, multi-trailer systems, railway systems, AGVs, and Autonomous lifting vehicles (ALVs) or by sea using barges. Similar with general transport systems, the goal of an efficient ITT system is to satisfy customer demands with minimal economical and environmental costs, such as energy consumption, traffic congestion, and green-house emissions. However, since ITT forms a complex network and a delay could adversely affect all following operations, punctuality is deemed as the most important criterion in ITT [21].

At present, ITT, e.g., in the port of Rotterdam, is implemented mainly by means of multi-trailer systems. Such systems use manned trucks, pulling trains of five trailers. The performance of using multi-container yard trucks, AGVs, and ALVs for ITT is simulated and compared [21] with real data from the port of Rotterdam. Automated vehicles (AGVs and ALVs) are found to be superior to manned vehicles in terms of punctuality. In [129], barges are also considered and integer programming models are proposed based on a time-space graph to optimize and analyze ITT performance. Barges in an ITT system are shown to be beneficial in terms of efficiency for delivering cargo and are even critical in instances with a large number of containers. Hybrid transport modes for ITT are recommended by the authors. Aiming at enhancing the management of real-time data and traffic flow of ITT, [40] proposes a cloud-based information system. The system relies on advanced data-collection and information exchange technologies and acts as a decision support system to the port community. A chronological overview of approaches, methods, and contributions in the area of ITT can be found in [41] and references therein. Research on ITT is there categorized as simulation, optimization, information system approaches, and case studies. Most of the work studies ITT scenarios in the port of Rotterdam, Maasvlakte I and II in particular.

2.5 Conclusions

In this chapter, relevant literature in view of the development of waterborne AGVs is reviewed. Two existing intelligent vehicles, land-based AGVs and intelligent marine surface vehicles, are discussed. The development trend, experiences, and technologies in using AGVs instead of manned trucks in automated container terminals well motivate the use of waterborne AGVs instead of manned barges in port areas. Research on intelligent marine surface vehicles has been mostly targeted for military or research purposes; transportation oriented design is rare. Safer, more sustainable, and efficient marine transport systems could be expected if more intelligent waterborne transport vehicles are used.

Research related to the focus of this thesis, coordination of waterborne AGVs, have been reviewed. There are several options of control techniques that could be applicable to motion control problems in general. Considering the advantages of MPC and the characteristics of waterborne AGVs with applications to ITT, MPC turns out to be the most suitable for the motion control problems of waterborne AGVs. Extensive research on both theoretical analysis and applications of MPC, distributed MPC, robust MPC, and RDMPC has been done. Special techniques still need to be developed for the particular cases of a single waterborne AGV, networked waterborne AGVs with cooperative distributed solutions, and

waterborne AGVs with environmental disturbances. Different scheduling approaches have also been discussed. The scheduling problem of waterborne AGVs can be formulated as a PDP. Special scheduling issues for waterborne AGVs in port areas are that emissions and coordinated berthing times are critical. Based on the literature review in this chapter, advanced coordinating strategies will be proposed for waterborne AGVs applied in typical ITT scenarios in Chapters 3 - 7.

Chapter 3

Dynamic models of waterborne AGVs

As discussed in Chapters 1 and 2, coordination strategies for waterborne AGVs are based on proper knowledge of the system dynamics. In this chapter, two types of mathematical models, i.e., simulation and prediction models, that describe dynamic waterborne AGV behaviors are proposed. Section 3.1 models the scenario of Inter Terminal Transport (ITT) using waterborne AGVs based on graphs. In Section 3.2, these models are approximated using a successive linearization approach for predicting system trajectories. Section 3.3 concludes this chapter. The models presented in this chapter will be used in the sequel throughout the thesis.

The research discussed in this chapter is partially based on [149, 150, 152].

3.1 Modeling of waterborne AGVs for ITT

In an ITT network using waterborne AGVs, a set of transportation tasks are assigned to a fleet of deployed waterborne AGVs. Each waterborne AGV is required to load/unload a certain number of containers at an origin terminal, depart at a specified time, travel along a reference route, arrive at a specified terminal at a specified time, and load/unload a certain number of containers. Waterborne AGVs are equipped on board with processing, measurement, and communication devices to measure their own system states, communicate with other waterborne AGVs within a certain range, and process locally. When waterborne AGVs will not collide over a finite number of predictable steps in any case, they maneuver independently to fulfill their assigned ITT tasks. However, when several waterborne AGVs are involved in a neighborhood, negotiations and alterations of trajectories to avoid collisions are typically necessary for ensuring the overall safety. In this section, waterborne AGVs for ITT are modeled using graphs [18] $\mathcal{G}(k) = (\mathcal{V}(k), \mathcal{E}(k))$ where k is the discrete time step and relates to continuous time t as $t = kT_s$ with T_s as the sampling time. The graph vertices $\mathcal{V}(k)$ representing waterborne AGVs are modeled in Section 3.1.1, and the graph edges $\mathcal{E}(k)$ representing couplings between waterborne AGVs are modeled in Section 3.1.2.

3.1.1 Waterborne AGV model

This section models the vertices $\mathcal{V}(k)$ in graph $\mathcal{G}(k)$, i.e., waterborne AGV dynamic models. Models of marine crafts with different degrees of freedom (DOFs) for different purposes have been elaborated on in [30]. Generally, a marine surface vehicle experiences motions in six DOFs, as shown in Figure 3.1a. For tracking problems of surface vehicles, models with three DOFs in the horizontal plane, as shown in Figure 3.1b, are sufficient to capture the main system characteristics [30] based on the assumption that the roll and pitch motions are small. Waterborne AGVs are modeled based on a three DOF maneuvering model in [30].

Consider that a set $\mathcal{V}(k) = \{1, 2, ..., n(k)\}$ of $n(k)$ numbered homogeneous waterborne AGVs and for each waterborne AGV $p \in \mathcal{V}(k)$, the dynamics that consider environmental disturbances due to wind, waves, and current are modeled as:

$$\dot{\boldsymbol{\eta}}_p(t) = \boldsymbol{R}(\psi_p(t))\boldsymbol{\nu}_p(t), \tag{3.1}$$

$$\begin{aligned}\dot{\boldsymbol{\nu}}_p(t) =& (\boldsymbol{M}_{\mathrm{RB}} + \boldsymbol{M}_{\mathrm{A}})^{-1}(\boldsymbol{\tau}_p(t) + \boldsymbol{R}^{\mathrm{T}}(\psi_p(t))\boldsymbol{R}^{\mathrm{T}}(\psi_b(t))b(t) \\ & - \left(\boldsymbol{C}_{p,\mathbf{RB}}(t) + \boldsymbol{C}_{p,\mathbf{A}}(t)\right)\boldsymbol{\nu}_p(t) + (\boldsymbol{D}_{\mathbf{L}} + \boldsymbol{D}_{p,\mathbf{NL}}(t))\boldsymbol{\nu}_p(t)),\end{aligned} \tag{3.2}$$

where $\begin{bmatrix} \boldsymbol{\eta}_p{}^{\mathrm{T}} & \boldsymbol{\nu}_p{}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{\tau}_p$ are system states and inputs, respectively, and

$$\boldsymbol{\eta}_p = \begin{bmatrix} x_p \\ y_p \\ \psi_p \end{bmatrix}, \ \boldsymbol{\nu}_p = \begin{bmatrix} u_p \\ v_p \\ r_p \end{bmatrix}, \ \boldsymbol{\tau}_p = \begin{bmatrix} \tau_{u,p} \\ \tau_{v,p} \\ \tau_{r,p} \end{bmatrix},$$

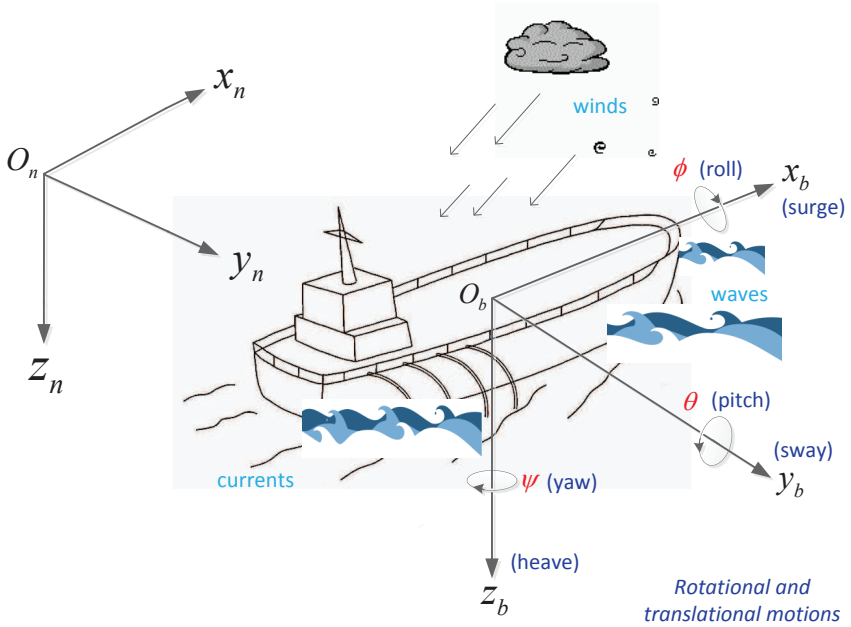
where x_p, y_p and ψ_p are coordinates, and heading angle, respectively, in the inertial coordinate $\{n\}$. Linear velocities in surge and sway are expressed in the body-fixed coordinate $\{b_p\}$ of waterborne AGV p as u_p and v_p, respectively, and the angular velocity of the heading angle is expressed by r_p. The control input vector for waterborne AGV p consists of surge force, sway force, and yaw moment represented by $\tau_{u,p}$, $\tau_{v,p}$ and $\tau_{r,p}$, respectively. Due to physical limitations such as maximum speeds, maximum engine power, etc., constraints on states and control inputs are usually imposed as:

$$\boldsymbol{\nu}_{p\min} \leqslant \boldsymbol{\nu}_p \leqslant \boldsymbol{\nu}_{p\max}, \ \boldsymbol{\tau}_{p\min} \leqslant \boldsymbol{\tau}_p \leqslant \boldsymbol{\tau}_{p\max}. \tag{3.3}$$

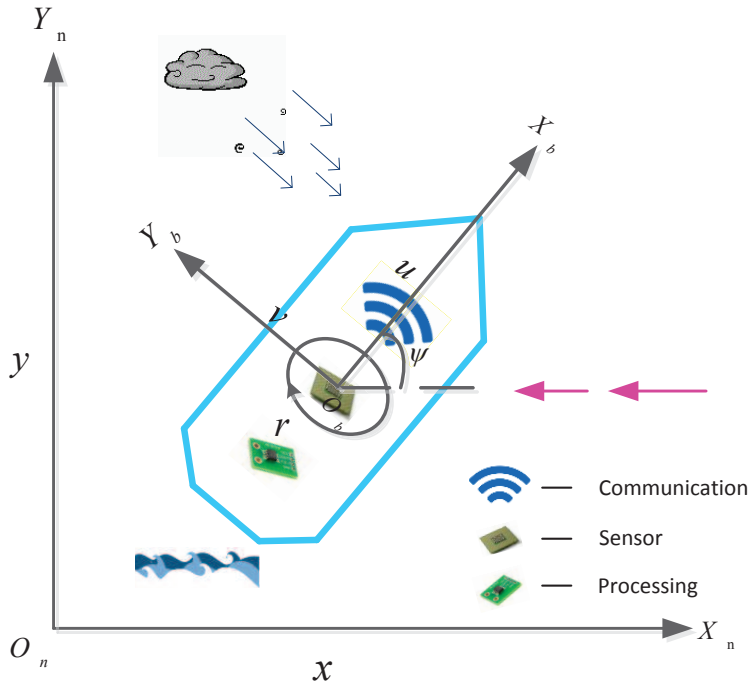
In (3.1), $\boldsymbol{R}(\psi_p)$ is a rotation matrix relating motions in $\{n\}$ and $\{b_p\}$, defined as:

$$\boldsymbol{R}(\psi_p) = \begin{bmatrix} \cos(\psi_p) & -\sin(\psi_p) & 0 \\ \sin(\psi_p) & \cos(\psi_p) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Environmental disturbances due to wind, waves, and current are modeled as a non-rotational force b with angle ψ_b in $\{n\}$. The effects of b along three DOFs are mapped by the rotation vector $\boldsymbol{R}^{\mathrm{T}}(\psi_b(t)) = \begin{bmatrix} \cos(\psi_b(t)) & \sin(\psi_b(t)) & 0 \end{bmatrix}^{\mathrm{T}}$ assuming that the force is acting on the gravity center. Disturbance effects in $\{n\}$ are further rotated to $\{b_p\}$ to account for the force changes with the heading of waterborne AGV p. Note that since all the waterborne AGVs in a port area experience the same environmental disturbances predicted by the port authority, the disturbance related parameters are without subscript $\bullet_p$.



(a) Marine surface vehicle in six DOFs.



(b) Waterborne AGV in three DOFs.

Figure 3.1: Modeling waterborne AGV dynamics

Rigid-body and added mass matrices are the same for all the homogeneous waterborne AGVs and are given as:

$$\boldsymbol{M}_{\mathrm{RB}} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & m x_{\mathrm{g}} \\ 0 & m x_{\mathrm{g}} & I_{\mathrm{z}} \end{bmatrix}, \boldsymbol{M}_{\mathrm{A}} = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix},$$

where subscripts $\cdot_{\mathrm{RB}}$ and $\cdot_{\mathrm{A}}$ stand for rigid body and added force related matrices, respectively; m is the mass of waterborne AGVs[1]; I_{z} is the moment of inertia in the yaw rotation; and x_{g} is the distance between the center of gravity of waterborne AGVs to the center of $\{b_p\}$. Similarly,

$$\boldsymbol{C}_{p,\mathrm{RB}}(t) = \begin{bmatrix} 0 & 0 & -m\left(x_{\mathrm{g}} r_p + v_p\right) \\ 0 & 0 & m u_p \\ m\left(x_{\mathrm{g}} r_p + v_p\right) & -m u_p & 0 \end{bmatrix},$$

$$\boldsymbol{C}_{p,\mathrm{A}}(t) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}} v_p + \left(N_{\dot{v}} + Y_{\dot{r}}\right) r_p/2 \\ 0 & 0 & -X_{\dot{u}} u_p \\ -Y_{\dot{v}} v_p - \left(N_{\dot{v}} + Y_{\dot{r}}\right) r_p/2 & X_{\dot{u}} u_p & 0 \end{bmatrix}$$

are rigid-body, and added Coriolis and centripetal matrices of waterborne AGV p, respectively.

Damping forces are separated into two parts: a linear part as

$$\boldsymbol{D}_{\mathrm{L}} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix},$$

and a nonlinear part as

$$\boldsymbol{D}_{p,\mathrm{NL}}(t) = \begin{bmatrix} -X_{|u|u}|u_p| - X_{uuu} {u_p}^2 & 0 & 0 \\ 0 & -Y_{|v|v}|v_p| - Y_{|r|v}|r_p| & -Y_{|v|r}|v_p| - Y_{|r|r}|r_p| \\ 0 & -N_{|v|v}|v_p| - N_{|r|v}|r_p| & -N_{|v|r}|v_p| - N_{|r|r}|r_p| \end{bmatrix}.$$

Hydrodynamic derivatives follow the notations in [123]. For instance, the hydrodynamic added mass force X along the x axis due to an acceleration $\dot{u}$ in the x direction is written as

$$X = -X_{\dot{u}}\dot{u}, \quad X_{\dot{u}} := \frac{\partial X}{\partial \dot{u}},$$

which implies $\{\boldsymbol{M}_{\mathrm{A}}\}_{11} = -X_{\dot{u}}$. Readers are referred to [123] for more details.

Depending on whether environmental disturbances are perfectly known beforehand or not, models (3.1) – (3.2) are further discussed as follows.

Waterborne AGV models with perfectly known disturbances

In a deterministic case, environmental disturbances are known perfectly beforehand. Particularly, we consider waterborne AGVs with known constant current which is non-rotational,

[1]For the controller design problems in Chapters 4 – 6, the empty waterborne AGV mass m is considered without any containers on board. In Chapter 7, the waterborne AGV mass is considered as the sum of the empty waterborne AGV mass m and the mass of all containers on board.

and has fixed speed V_c and angle β_c in $\{n\}$ ($\dot{V}_\mathrm{c}=0$ and $\dot{\beta}_\mathrm{c}=0$). A rotation to $\{b_p\}$ is

$$\boldsymbol{\nu}_{p,\mathrm{c}} = \boldsymbol{R}^{\mathrm{T}}(\psi_p)\begin{bmatrix} V_\mathrm{c}\cos\beta_\mathrm{c} \\ V_\mathrm{c}\sin\beta_\mathrm{c} \\ 0 \end{bmatrix}. \tag{3.4}$$

The influences of current are then expressed by the relative velocity in $\{b_p\}$ between the waterborne AGV hull and the fluid as $\boldsymbol{\nu}_{p,\mathrm{r}}(t) = \boldsymbol{\nu}_p(t) - \boldsymbol{\nu}_{\mathrm{c},p}(t) = \begin{bmatrix} u_{p,\mathrm{r}}(t) & v_{p,\mathrm{r}}(t) & r(t) \end{bmatrix}^{\mathrm{T}}$, and satisfy:

$$\boldsymbol{M}_{\mathrm{RB}}\dot{\boldsymbol{\nu}}_p(t) + \boldsymbol{M}_{\mathrm{A}}\dot{\boldsymbol{\nu}}_{p,\mathrm{r}}(t) + \boldsymbol{C}_{p,\mathrm{RB}}(t)\boldsymbol{\nu}_p(t) + (\boldsymbol{C}_{p,\mathrm{A}}(\boldsymbol{\nu}_{p,\mathrm{r}}) + \boldsymbol{D}_{\mathbf{L}} + \boldsymbol{D}_{p,\mathrm{NL}}(\boldsymbol{\nu}_{p,\mathrm{r}}))\boldsymbol{\nu}_{p,\mathrm{r}}(t) = \boldsymbol{\tau}_p.$$

Since $\boldsymbol{R}$ satisfies $\frac{\mathrm{d}}{\mathrm{d}t}\{\boldsymbol{R}(\psi_p)\} = r_p\boldsymbol{R}(\psi_p)\boldsymbol{S}$ with

$$\boldsymbol{S} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

we have $\dot{\boldsymbol{\nu}}_{p,\mathrm{r}} = \dot{\boldsymbol{\nu}}_p - r_p\boldsymbol{S}^{\mathrm{T}}\boldsymbol{\nu}_{p,\mathrm{c}}$. Then, (3.2) can be rewritten as:

$$\begin{aligned}\dot{\boldsymbol{\nu}}_p(t) =& (\boldsymbol{M}_{\mathrm{RB}} + \boldsymbol{M}_{\mathrm{A}})^{-1}\left(\boldsymbol{\tau}_p(t) - (\boldsymbol{C}_{p,\mathrm{A}}(\boldsymbol{\nu}_{p,\mathrm{r}}) + \boldsymbol{D}_{\mathbf{L}} + \boldsymbol{D}_{p,\mathrm{NL}}(\boldsymbol{\nu}_{p,\mathrm{r}}))\boldsymbol{\nu}_{p,\mathrm{r}}(t)\right. \\ & \left. -\boldsymbol{C}_{\mathrm{RB}}(t)\boldsymbol{\nu}_p(t) + \boldsymbol{M}_{\mathrm{A}}r_p(t)\boldsymbol{S}^{\mathrm{T}}\boldsymbol{\nu}_{\mathrm{c},p}(t)\right),\end{aligned} \tag{3.5}$$

where the added Coriolis and centripetal matrix depends on $\boldsymbol{\nu}_{p,\mathrm{r}}$ as

$$\boldsymbol{C}_{p,\mathrm{A}}(\boldsymbol{\nu}_\mathrm{r}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v_{p,r} + (N_{\dot{v}} + Y_{\dot{r}})\,r_p/2 \\ 0 & 0 & -X_{\dot{u}}u_{p,r} \\ -Y_{\dot{v}}v_{p,r} - (N_{\dot{v}} + Y_{\dot{r}})\,r_p/2 & X_{\dot{u}}u_{p,r} & 0 \end{bmatrix},$$

and the nonlinear damping matrix depends on $\boldsymbol{\nu}_{p,\mathrm{r}}$ as

$$\boldsymbol{D}_{p,\mathrm{NL}}(\boldsymbol{\nu}_r) = \begin{bmatrix} -X_{|u|u}\,|u_{p,r}| - X_{uuu}{u_{p,r}}^2 & 0 & 0 \\ 0 & -Y_{|v|v}\,|v_{p,r}| - Y_{|r|v}\,|r_p| & -Y_{|v|r}\,|v_{p,r}| - Y_{|r|r}\,|r_p| \\ 0 & -N_{|v|v}\,|v_{p,r}| - N_{|r|v}\,|r_p| & -N_{|v|r}\,|v_{p,r}| - N_{|r|r}\,|r_p| \end{bmatrix}.$$

Waterborne AGV models with not perfectly known disturbances

Generally, waterborne AGVs travel in good weather based on roughly predicted values of b and ψ_b. For example, the port authority cooperates with local water management organizations and installs hydrological and meteorological sensors at different locations in the port of Rotterdam to provide information on visibility, tides, flow rates, wave heights, wind speeds and directions [103]. However, predictions based on weather forecast and sensed information are mostly stochastically uncertain. Therefore, it is assumed that environmental disturbances b and ψ_b could in principle be obtained from the port authority's meteorological predictions, but with the existence of prediction uncertainties following a normal distribution on the magnitude of disturbance forces b as, i.e.,

$$b \sim N(\bar{b}, \Sigma), \tag{3.6}$$

where the mean $\bar{b}$ is predicted by the port authority and the covariance Σ reflects the prediction accuracy. Similar use and assumptions of weather prediction information have been made for energy-efficient building systems [94].

Summarizing, waterborne AGVs $p \in \mathcal{V}$ maneuver independently to fulfill assigned ITT tasks when far away. The dynamics are defined as (3.1), (3.5) when environmental disturbances are perfectly known, and as (3.1) – (3.2), (3.6) when environmental disturbances are not perfectly known. In both cases, system physical limits (3.3) need to be satisfied.

3.1.2 Network model

This section models the edges $\mathcal{E}(k)$ in graph $\mathcal{G}(k)$, i.e., the couplings between waterborne AGVs. When waterborne AGVs are within each other's communication range $D_{\mathbf{c}}$, shown as the black dotted circle in Figure 3.2a, couplings arise as pairwise collision avoidance constraints to enforce a minimal safety distance $D_{\mathbf{s}}$ (red dashed circle). Note that in real implementations, $D_{\mathbf{c}}$ and $D_{\mathbf{s}}$ that are related to the ship domain [100] are usually not circular. However, the determination of the shape and size of a ship domain is out of the scope of this thesis. The algorithms designed in the latter chapters based on circles with constant radii are for simplicity and are also applicable to cases where non-circular ship domains with varying sizes are involved. Based on Figure 3.2a, we first define a subset $\mathcal{V}_{\mathrm{w}}(k) \subseteq \mathcal{V}$ of $n_{\mathrm{w}}(k)$ working waterborne AGVs. A waterborne AGV $p \in \mathcal{V}_{\mathrm{w}}(k)$ if and only if its assigned departure time has been reached and it has not arrived at its assigned destination yet, i.e.,

$$t \geqslant t_p,\ \|\mathbf{r}_p(k) - \boldsymbol{d}_p\|_2 \geqslant d_{\mathrm{tol}}, \tag{3.7}$$

where $\|\cdot\|_2$ denotes the two-norm Euclidian distance and $\mathbf{r}_p(k) = \begin{bmatrix} x_p(k) & y_p(k) \end{bmatrix}^{\mathrm{T}}$ is the measured position of waterborne AGVs p; t_p and $\boldsymbol{d}_p$ are the assigned departure time and destination, respectively; $d_{\mathrm{tol}} > 0$ is a small tolerance.

The edge set is then defined as $\mathcal{E}(k) = \left\{ e_{p,q}(k) | p,q \in \mathcal{V}_{\mathbf{w}}(k), p < q \right\}$ with

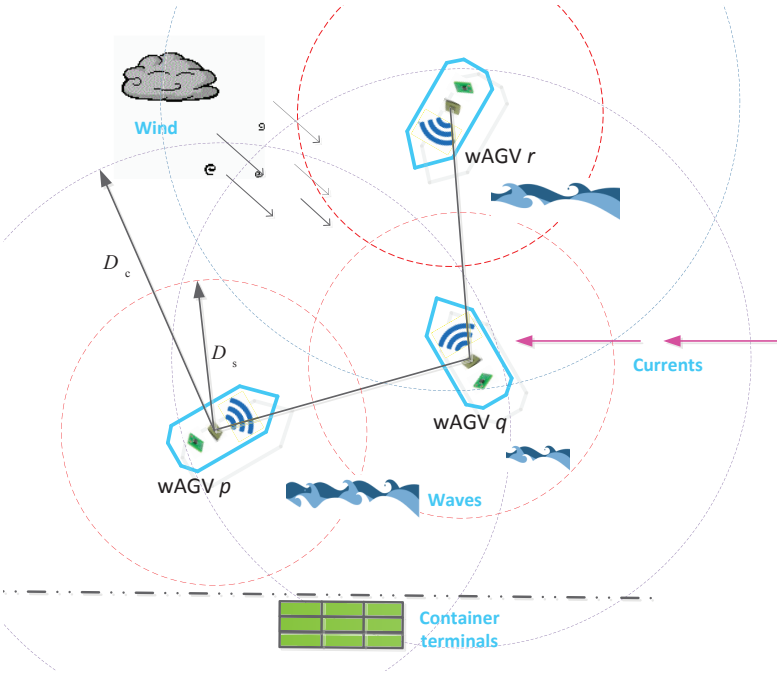
$$e_{p,q}(k) = \begin{cases} 1, & \text{if } d_{p,q}(k) \leqslant D_{\mathbf{c}}, \\ 0, & \text{otherwise}, \end{cases}$$

where $D_{\mathbf{c}} = 2u_{\max}T_s N_{\mathrm{p}} + l + \alpha$ with prediction horizon N_{p}; $u_{\max}$ is the maximum surge speed of waterborne AGVs; l is one waterborne AGV length; α is the margin for uncertainties; and $d_{p,q}(k) = \left\| \boldsymbol{r}_p(k) - \boldsymbol{r}_q(k) \right\|_2$ is the Euclidean distance between waterborne AGV p and q. Collision avoidance couplings are then imposed to pairwise waterborne AGVs $\left\{ (p,q) | e_{p,q}(k) = 1, e_{p,q}(k) \in \mathcal{E}(k) \right\}$ as:

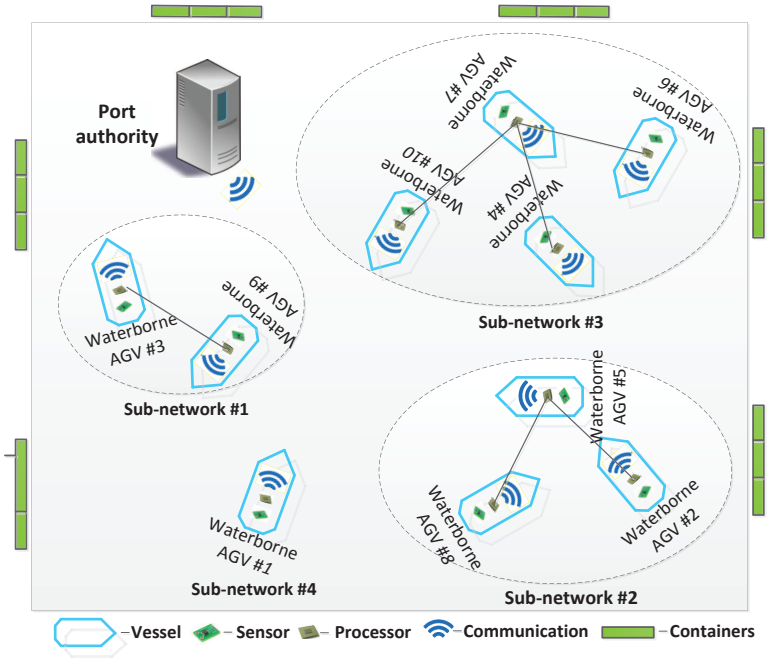
$$d_{p,q}(k) = \left\| \boldsymbol{r}_p(k) - \boldsymbol{r}_q(k) \right\|_2 \geqslant D_{\mathrm{s}}. \tag{3.8}$$

The above logic implies that collision avoidance couplings emerge when there is a possibility of collision over the next prediction horizon if two waterborne AGVs are sailing at their maximum speeds. This is a relatively conservative logic but guarantees safety if proper actions are taken. Coupling constraints (3.8) hinder independent computations of waterborne AGVs.

It is usually the case that we have groups of waterborne AGVs that are coupled within a group but are decoupled between groups, as Figure 3.2b shows. In this case, $\mathcal{G}(k)$ is



(a) Waterborne AGV couplings; communication (black dotted) and safety (red dashed) ranges.



(b) Groups of waterborne AGVs: coupled intra groups and decoupled inter groups.

Figure 3.2: Waterborne AVG networks for ITT.

disconnected [18]. Connected components that are subgraphs of $\mathcal{G}(k)$ can be computed using algorithms like breadth-first search [45] based on $\mathcal{E}(k)$. Let $n_{\mathcal{G}}(k)$ subgraphs $\mathcal{G}_s(k) = (\mathcal{V}_s(k), \mathcal{E}_s(k))$ for $s = 1,2,..,n_{\mathcal{G}}(k)$ be derived, where $\mathcal{V}_s(k)$ is the subset of $n_s(k)$ waterborne AGVs and $\mathcal{E}_s(k)$ defines the set of collision avoidance coupling pairs in $\mathcal{G}_s(k)$. Obviously, $\mathcal{V}_s(k) \subseteq \mathcal{V}_{\mathrm{w}}(k)$ and $1 \leqslant n_s(k) = \left|\mathcal{V}_s(k)\right| \leqslant n_{\mathrm{w}}(k) \leqslant n(k)$. All the numbered working waterborne AGVs appear once and only once in one of the subgraphs, i.e., $\cup_{s=1}^{n_{\mathcal{G}}(k)} \mathcal{V}_s(k) = \mathcal{V}_{\mathrm{w}}(k)$, $\cap_{s=1}^{n_{\mathcal{G}}(k)} \mathcal{V}_s(k) = \emptyset$ and $\sum_{s=1}^{n_{\mathcal{G}}(k)} n_s(k) = n_{\mathrm{w}}(k)$. At each time step k, subgraphs $\mathcal{G}_s(k) = (\mathcal{V}_s(k), \mathcal{E}_s(k))$ are constructed and assumed to be constant over the next prediction horizon. Cases of $\mathcal{G}_s(k)$ include:

1. Subgraphs $\mathcal{G}_s(k), s = 1,...,n_{\mathcal{G}}(k)$ are singleton graphs when $n_s(k) = 1$ and $\mathcal{E}_s(k) = \mathbf{0}$;
2. There are multiple subgraphs $\mathcal{G}_s(k)$ with $\mathcal{E}_s(k) \neq \mathbf{0}$ and $n_s(k) \geqslant 1$; and
3. There is only one subgraph $\mathcal{G}_s(k)$ in $\mathcal{G}(k)$ when $\mathcal{G}_s(k) = \mathcal{G}(k)$ and $n_s(k) = n_{\mathrm{w}}(k)$.

The first case corresponds to the problem for one single waterborne AGV considered in Chapter 4. The second case corresponds to the problem for multiple waterborne AGVs with multiple couplings that are considered in Chapters 5 and 7. The third case corresponds to the problem for multiple waterborne AGVs considered in Chapter 6. All these problems can be modeled and will be solved based on the above constructed graph structures.

3.2 Successively linearized models

As discussed in Chapter 2, one of the issues concerning model predictive control (MPC) is that optimization problems need to be solved online repetitively. The complexity of an MPC controller with certain controller parameters mainly depends on the characteristics (order, nonlinearities) of the prediction model it uses and imposed system constraints. For highly nonlinear systems with nonlinear constraints, a dilemma is usually faced by MPC. On the one hand, stringent demands on system performance generally require a high accuracy prediction model. On the other hand, online prediction and optimization of future system behavior based on complex prediction models is too time consuming, which is undesirable for real-time applications. A compromise between optimality and computational complexity has to be made for fast and nonlinear waterborne AGV dynamics.

Dynamic models of waterborne AGVs as presented in Section 3.1 with nonlinear dynamics and non-convex collision avoidance constraints are not preferable directly for controller design. The reasons are twofold: *1*) they are computationally too complex for real-time control; and *2*) distributed and robust control design are difficult if not impossible. A comparison on performance of using nonlinear and single step linearized models of waterborne AGVs in MPC has been made in [146]. It was found that computational time is much longer using nonlinear prediction models than using linearized models. In this section, we introduce a successive linearization approach and derive approximated dynamic waterborne AGV models and constraints for MPC controller design. Details on MPC controllers will be presented in the later chapters.

The basic idea is to utilize the whole sequence of control inputs from a previous MPC step and pre-calculate a shifted system trajectory for linearizations over all prediction steps at the next MPC step. We generalize the dynamics (3.1) – (3.2) of waterborne AGV p as:

$$\dot{\boldsymbol{x}}_p(t) = f(\boldsymbol{x}_p(t), \boldsymbol{u}_p(t), b(t)), \tag{3.9}$$

where $f: \mathbb{R}^6 \times \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^6$ is a nonlinear smooth function with system states $\boldsymbol{x}_p = \begin{bmatrix} \boldsymbol{\eta}_p^T & \boldsymbol{\nu}_p^T \end{bmatrix}^T$ and control inputs $\boldsymbol{u}_p = \boldsymbol{\tau}_p$. For numerical simulations, the continuous time model (3.9) is discretized with the zero-order-hold assumption as:

$$\boldsymbol{x}_p(k+1) = \boldsymbol{x}_p(k) + \int_{k\mathrm{T}_s}^{(k+1)\mathrm{T}_s} f(\boldsymbol{x}_p(k), \boldsymbol{u}_p(k), b(k))\mathrm{d}t. \tag{3.10}$$

Then at each time step k, for each waterborne AGV $p \in \mathcal{V}(k)$, successive linearizations are implemented as the following three steps:

1. Obtain seed [51] input trajectory $\boldsymbol{u}_p^0(i|k)$ whereby, $(i|k)$ denotes the ith prediction step at time step k, and the superscript $\bullet^0$ denotes seed trajectories. Whenever contextually clear, prediction step i for control/disturbance inputs is over $0, 1, ..., N_\mathrm{p} - 1$ and for states over $0, 1, ..., N_\mathrm{p}$. Consider the previous time step $k-1$ ($k > 1$), the calculated optimal control input sequence is $\boldsymbol{u}_p(i|k-1)$. Conventionally, the first element $\boldsymbol{u}_p(0|k-1)$ is applied to the system and the rest are disregarded. For linearizations at step k, we make extensive use of the "tail" which is shifted as:

   $$\boldsymbol{u}_p^0(i|k) = \boldsymbol{u}_p(i|k-1) \tag{3.11}$$

   for $i = 0, 1, ..., N_\mathrm{p} - 2$ and

   $$\boldsymbol{u}_p^0(N_\mathrm{p} - 1|k) = \boldsymbol{u}_p(N_\mathrm{p} - 1|k-1). \tag{3.12}$$

2. Obtain seed state trajectory $\boldsymbol{x}_p^0(i|k)$. With an initial state $\boldsymbol{x}_p^0(k|k) = \boldsymbol{x}_p(k)$ and $\boldsymbol{u}_p^0(i|k)$, apply $\boldsymbol{u}_p^0(i|k)$ to (3.10) iteratively or to (3.9) using available ordinary differential equation solvers (e.g., in MATLAB [75]) which provide higher precision than (3.10). This is straightforward when disturbances are perfectly known. However, when disturbances are not perfectly known as modeled in (3.6), the values of $b(i|k)$ remain unknown. Therefore, we define a seed disturbance input trajectory as $b^0(i|k) = \bar{b}(i|k)$ by utilizing the predicted mean values from the port authority. One of the justifications for this definition is that deviations of the real value $b(i|k)$ from $b^0(i|k)$ (or $\bar{b}(i|k)$) are small for a high probability, and small deviations satisfy the assumption of Jacobian linearizations conducted at the next step. Besides, the widely known extended Kalman filter [48] also conducts linearizations of nonlinear systems about mean values of random variables.

3. Linearize nonlinear dynamics and constraints at seed trajectory $\left(\boldsymbol{x}_p^0(i|k), \boldsymbol{u}_p^0(i|k)\right)$ in a deterministic case or $\left(\boldsymbol{x}_p^0(i|k), \boldsymbol{u}_p^0(i|k), b^0(i|k)\right)$ in an uncertain case. Define small perturbations around the seed trajectory as $(\Delta\boldsymbol{x}_p(i|k), \Delta\boldsymbol{u}_p(i|k), \Delta b(i|k))$ ($\Delta b(i|k) = \mathbf{0}$ in a deterministic case) which satisfy:

   $$\boldsymbol{x}_p(i|k) = \boldsymbol{x}_p^0(i|k) + \Delta\boldsymbol{x}_p(i|k), \tag{3.13}$$

   $$\boldsymbol{u}_p(i|k) = \boldsymbol{u}_p^0(i|k) + \Delta\boldsymbol{u}_p(i|k), \tag{3.14}$$

   $$b(i|k) = b^0(i|k) + \Delta b(i|k). \tag{3.15}$$

Substituting (3.13) – (3.15) into (3.10), we get

$$\boldsymbol{x}_p^0(i+1|k)+\Delta\boldsymbol{x}_p(i+1|k)=\boldsymbol{x}_p^0(i|k)+\Delta\boldsymbol{x}_p(i|k)+$$
$$\int_{k\mathrm{T}_s}^{(k+1)\mathrm{T}_s}\boldsymbol{f}\left(\boldsymbol{x}_p^0(i|k)+\Delta\boldsymbol{x}_p(i|k),\boldsymbol{u}_p^0(i|k)+\Delta\boldsymbol{u}_p(i|k),b^0(i|k)+\Delta b(i|k)\right)\mathrm{d}t. \tag{3.16}$$

The integrator term, by applying Taylor's theorem and neglecting the higher order terms than the first order, is approximated as:

$$\int_{k\mathrm{T}_s}^{(k+1)\mathrm{T}_s}\boldsymbol{f}\left(\boldsymbol{x}_p^0(i|k),\boldsymbol{u}_p^0(i|k),b^0(i|k)\right)\mathrm{d}t+$$
$$\int_{k\mathrm{T}_s}^{(k+1)\mathrm{T}_s}\boldsymbol{A}_p^{\mathrm{c}}(i|k)\Delta\boldsymbol{x}_p(i|k)+\boldsymbol{B}_p^{\mathrm{c}}(i|k)\Delta\boldsymbol{u}_p(i|k)+\boldsymbol{E}_p^{\mathrm{c}}(i|k)\Delta b(i|k)\mathrm{d}t \tag{3.17}$$

where

$$\boldsymbol{A}_p^{\mathrm{c}}(i|k)=\left.\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}\right|_{\left(\boldsymbol{x}_p^0(i|k),\boldsymbol{u}_p^0(i|k),b^0(i|k)\right)},$$
$$\boldsymbol{B}_p^{\mathrm{c}}(i|k)=\left.\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{u}}\right|_{\left(\boldsymbol{x}_p^0(i|k),\boldsymbol{u}_p^0(i|k),b^0(i|k)\right)},$$
$$\boldsymbol{E}_p^{\mathrm{c}}(i|k)=\left.\frac{\partial\boldsymbol{f}}{\partial b}\right|_{\left(\boldsymbol{x}_p^0(i|k),\boldsymbol{u}_p^0(i|k),b^0(i|k)\right)},$$

are continuous Jacobian state, input, and disturbance matrices, respectively. Then by (3.16) and (3.17), we reach the discrete linearized incremental model

$$\Delta\boldsymbol{x}_p(i+1|k)=\boldsymbol{A}_p^{\mathrm{d}}(i|k)\Delta\boldsymbol{x}_p(i|k)+\boldsymbol{B}_p^{\mathrm{d}}(i|k)\Delta\boldsymbol{u}_p(i|k)+\boldsymbol{E}_p^{\mathrm{d}}(i|k)\Delta b(i|k), \tag{3.18}$$

where $\boldsymbol{A}_p^{\mathrm{d}}(i|k)$, $\boldsymbol{B}_p^{\mathrm{d}}(i|k)$, and $\boldsymbol{E}_p^{\mathrm{d}}(i|k)$ are corresponding discrete Jacobian matrices.

In a similar way, non-convex collision avoidance constraints (3.8) are approximated as:

$$d_{p,q}^0(i|k)+\boldsymbol{C}(i|k)\Delta\boldsymbol{r}_p(i|k)+\boldsymbol{D}(i|k)\Delta\boldsymbol{r}_q(i|k)\geqslant D_{\mathrm{s}}, \tag{3.19}$$

where $\boldsymbol{C}(i|k)\in\mathcal{R}^{1\times 2}$ and $\boldsymbol{D}(i|k)\in\mathcal{R}^{1\times 2}$ are Jacobian matrices of function d with respect to $\boldsymbol{r}_p$ and $\boldsymbol{r}_q$ evaluated at $\left(\boldsymbol{r}_p^0,\boldsymbol{r}_q^0\right)$, respectively.

Time-varying but linear dynamic models (3.13) – (3.15), (3.18) and convex constraints (3.19) are then used to approximate original nonlinear dynamics (3.1) – (3.2) and non-convex collision avoidance constraints (3.8) for later controller design.

3.3 Conclusions

In this chapter, we have presented two types of dynamic models that describe the waterborne AGV behavior. The first type is for simulation purposes and models waterborne AGVs for ITT as graphs. Graph vertices represent waterborne AGV nonlinear dynamics and graph edges represent couplings between waterborne AGVs. These models are able to represent

all the typical ITT scenarios considered in the later chapters. However, waterborne AGV nonlinear dynamics and non-convex couplings are too complex for real-time applications and controller design. The second type approximates the first type using successive linearization in the framework of MPC. Linearized models could be beneficial by providing a trade-off among optimality, computational performance, and the ease of controller design. The waterborne AGV models used in Chapters 4 – 7 are all based on models presented in this chapter.

Chapter 4

Predictive path following with arrival time awareness

Regarding the Key Research Question on performance criteria for one waterborne AGV in Chapter 1, this chapter considers a single waterborne AGV controlled to carry out an assigned Inter Terminal Transport (ITT) task. A predictive path following with arrival time awareness controller is proposed based on the waterborne AGV dynamic models in Chapter 3. The controller is also able to handle static obstacles and avoid overshoots during switching of reference path segments.

The research discussed in this chapter has been published in [150].

4.1 Introduction

Currently, container movements are handled by land-based AGVs inside container terminals and by manned trucks between terminals. Waterborne AGVs are proposed as an alternative and innovative way for ITT with advantages presented in Chapter 1. A fundamental scenario in which a single waterborne AGV autonomously fulfills one assigned ITT task in a deterministic case is considered in this chapter. The waterborne AGV departs from an origin terminal, arrives at a destination terminal at specified times, and moves along a specified route that has been designed connecting these two terminals over water. The reference route is determined by waypoints and consists of straight line segments. In navigation and guidance systems for aircraft, waypoints are also often necessarily available to generate control references that are able to handle overshoot issues [56]. For marine vehicles, the reference generation approach, Line-of-Sight (LOS) [28], is usually applied in tracking routes defined by waypoints. However, overshoots during switching of line segments are seen in LOS based approaches. Besides, since a low rate of "non-performance" which happens when delays exist, is the most important criterion of ITT, the geometric time-independent reference route needs to be tracked while keeping the arrival time in mind. Moreover, the dynamics of a waterborne AGV are typically constrained multi-input multi-output systems, as seen in Chapter 3. Few control techniques can handle timing, overshoots, system constraints, and optimizing system performance quantitatively in a systematic way, as discussed

in Chapter 2.

In this chapter, based on the literature review in Chapter 2 and the models presented in Chapter 3, a predictive path following with arrival time awareness (PPF-ATA) controller is proposed based on model predictive control (MPC) that achieves for a single waterborne AGV that:

1. a given geometric reference path is tracked with deviations as small as possible;
2. a given arrival time requirement is met when a preferable time is feasible considering system limitations, or a minimal delay with respect to the preferable time within a specified time window otherwise; and
3. the aforementioned two design requirements are achieved in a speed-dependent energy economical way.

In particular, connected coordinate systems are established in which system kinematics are re-modeled. The benefits of doing so are twofold: first, tracking errors can be formulated more compactly; secondly, the along-track state is utilized in a reference switching logic so that overshoots are avoided. The switching logic combined with a coordinate transformation renders a continuous model in one coordinate system still applicable for successive linearizations. Moreover, a two-level double integrator model for parameterizing reference paths is proposed to achieve smooth tracking and arrival time awareness. The lower level is embedded in online MPC optimizations for smooth tracking. The higher level solves an optimal control problem considering distance-to-go and time-to-go each time step. Simulation results of two industrially relevant ITT case studies in the port of Rotterdam illustrate the effectiveness of the proposed model and control design for a waterborne AGV.

The remainder of this chapter is organized as follows. In Section 4.2, the connected path coordinate systems are established in which waterborne AGV kinematics are re-modeled for path following. The two-level double integrator model for path parameterization is proposed in Section 4.3. Section 4.4 introduces a switching logic with binary decision variables and coordinate transformations. The PPF-ATA algorithm is summarized in Section 4.5. Then in Section 4.6, simulation experiments and results are presented, followed by the concluding remarks of this chapter in Section 4.7.

4.2 Modeling in path coordinate systems

For controlling the waterborne AGV to move along the reference route of straight line segments, two system design requirements are first distinguished here: minimizing the *cross-track error* which is defined as the distance from the waterborne AGV's current position to the reference line, and minimizing the *along-track error*, defined as the error between the orthogonal projection point of the waterborne AGV on the reference line and an reference along path point. Besides the inertial coordinate system $\{n\}$ and the body-fixed coordinate system $\{b_p\}$ for modeling waterborne AGVs as in Chapter 3, connected path coordinate systems are established as shown in Figure 4.1. By re-modeling waterborne AGV kinematics (3.1) in path coordinate systems, the cross-track and along-track errors are conveniently formulated as linear functions of system states. Apart from that, the along-track state can be taken advantage of in a switching logic to avoid overshoots.

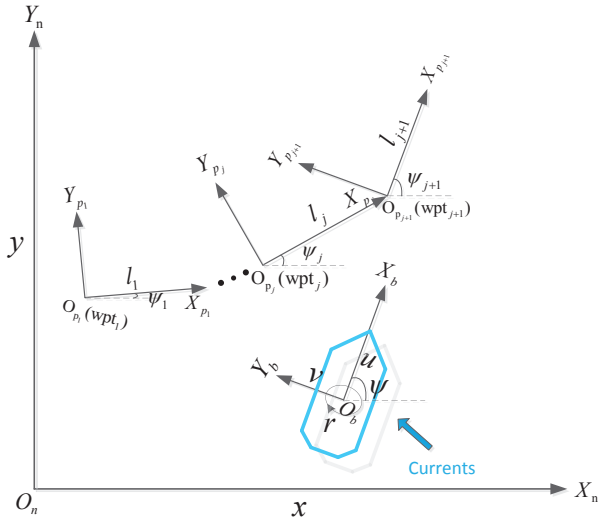


Figure 4.1: Re-modeling waterborne AGVs in path coordinate systems.

In Figure 4.1, the path coordinate systems $\{p_j\}$ $(j = 1,2,...)$ are based on connected reference paths; X_{p_j} is along the reference path and Y_{p_j} is vertical to the reference path pointing $\pi/2$ counterclockwise; O_{p_j} is the origin of the jth path coordinate system located at the jth waypoint connecting reference path $j-1$ and j. Lengths and angles with respect to X_{n} of reference path j are denoted as l_j and ψ_j, respectively. Then kinematics (3.1) [1] are modeled in path coordinate system $\{p_j\}$ as:

$$\dot{\boldsymbol{\eta}}_{\mathrm{p}_j}(t) = \boldsymbol{R}(\psi_{\mathrm{p}_j}(t))\boldsymbol{\nu}(t), \tag{4.1}$$

where, likewise, $\boldsymbol{\eta}_{\mathrm{p}_j} = \begin{bmatrix} x_{\mathrm{p}_j} & y_{\mathrm{p}_j} & \psi_{\mathrm{p}_j} \end{bmatrix}^{\mathrm{T}}$ is the pose expressed in $\{p_j\}$ with $\psi_{\mathrm{p}_j} = \psi - \psi_j$. $\boldsymbol{R}(\psi_{\mathrm{p}_j})$ is a rotation matrix relating motions between coordinate systems $\{p_j\}$ and $\{b\}$ and defined as:

$$\boldsymbol{R}(\psi_{\mathrm{p}_j}(t)) = \begin{bmatrix} \cos(\psi_{\mathrm{p}_j}) & -\sin(\psi_{\mathrm{p}_j}) & 0 \\ \sin(\psi_{\mathrm{p}_j}) & \cos(\psi_{\mathrm{p}_j}) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Kinetics of the waterborne AGV system are still expressed in frame $\{b\}$ as (3.2). Since waterborne AGV heading angle ψ is involved in (3.2) to transform current dynamics from $\{n\}$ to $\{b\}$, a state vector in $\mathcal{R}^7$ is defined as $\begin{bmatrix} \boldsymbol{\eta}_{\mathrm{p}_j}^{\mathrm{T}} & \psi & \boldsymbol{\nu}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$.

The new kinematics (4.1) are continuous within one coordinate system. However, the continuity is lost during the switch of coordinate systems. A transformation including rotation and translation of coordinates is then necessary. More specifically, we consider a switch from $\{p_j\}$ to $\{p_{j+1}\}$, as shown in Figure 4.1. The angle error between the new 'x' axis $X_{\mathrm{p}_{j+1}}$ and the old 'x' axis X_{p_j} is $\psi_{j+1} - \psi_j$. Since $\{p_j\}$ and $\{p_{j+1}\}$ are connected,

[1] Since only a single waterborne AGV is considered, the subscript $\bullet_p$ indicating waterborne AGV p is dropped in this chapter. A normal font subscript $\bullet_{\mathrm{p}}$ indicates variables in the path coordinate systems.

the new origin $O_{\mathrm{p}_{j+1}}$ has coordinates $(l_j, 0)$ relative to the old coordinate system $\{p_j\}$. A transformation of coordinates from $\{p_j\}$ to $\{p_{j+1}\}$ would then be:

$$\begin{bmatrix} x_{\mathrm{p}_{j+1}} \\ y_{\mathrm{p}_{j+1}} \end{bmatrix} = \begin{bmatrix} \cos(\psi_{j+1}-\psi_j) & \sin(\psi_{j+1}-\psi_j) \\ -\sin(\psi_{j+1}-\psi_j) & \cos(\psi_{j+1}-\psi_j) \end{bmatrix} \begin{bmatrix} x_{\mathrm{p}_j}-l_j \\ y_{\mathrm{p}_j}-0 \end{bmatrix}. \tag{4.2}$$

Meanwhile, a transformation of the heading angle from $\{p_j\}$ to $\{p_{j+1}\}$ would be:

$$\psi_{\mathrm{p}_{j+1}} = \psi_{\mathrm{p}_j} + \psi_j - \psi_{j+1}, \tag{4.3}$$

or simply as:

$$\psi_{\mathrm{p}_{j+1}} = \psi - \psi_{j+1}. \tag{4.4}$$

To obtain initial path coordinate states, measured states η need to be transformed from $\{n\}$ to $\{p_j\}$. Similar with (4.2), a transformation of position is:

$$\begin{bmatrix} x_{\mathrm{p}_j} \\ y_{\mathrm{p}_j} \end{bmatrix} = \begin{bmatrix} \cos(\psi_j) & \sin(\psi_j) \\ -\sin(\psi_j) & \cos(\psi_j) \end{bmatrix} \begin{bmatrix} x-x_{\mathrm{wpt}_j} \\ y-y_{\mathrm{wpt}_j} \end{bmatrix}, \tag{4.5}$$

where $\left(x_{\mathrm{wpt}_j}, y_{\mathrm{wpt}_j}\right)$ is the coordinate of waypoint j, or origin of $\{p_j\}$ in $\{n\}$. The transformation of heading angle is then the same with (4.4). So far, we have re-modeled kinematics (4.1) with coordinate transformations of (4.2) and (4.4) for transforming coordinates between $\{p_j\}$ and $\{p_{j+1}\}$, and (4.4) and (4.5) for transforming coordinates between $\{n\}$ and $\{p_j\}$.

4.3 Two-level double integrator dynamics

For an ITT task, the reference path is given as a sequence of time-independent straight-line segments, i.e., a path following rather than a trajectory tracking problem needs to be solved. To obtain smoothly tractable along-track references and at the same time satisfying the timing requirements, the geometric path is parameterized by double-integrator dynamics modeled at two levels, as shown in Figure 4.2. The lower level is embedded in online MPC optimizations. The higher level solves an optimal control problem considering distance-to-go and time-to-go which are fed back from the lower level double-integrator dynamics and provides the lower level timing aware references over the next receding prediction horizon.

The double integrator dynamics are modeled as:

$$\boldsymbol{x}_{\mathrm{s}}(k+1) = \boldsymbol{A}_{\mathrm{s}}\boldsymbol{x}_{\mathrm{s}}(k) + \boldsymbol{B}_{\mathrm{s}}u_{\mathrm{s}}(k), \tag{4.6}$$

where $\boldsymbol{x}_{\mathrm{s}} = \begin{bmatrix} s & v_{\mathrm{s}} \end{bmatrix}^{\mathrm{T}}$ and $u_{\mathrm{s}} = a_{\mathrm{s}}$. The scalar s is introduced as the traveled distance of the waterborne AGV along the straight-line reference paths with v_{s} as its velocity and a_{s} as its acceleration. State and input matrices are:

$$\boldsymbol{A}_{\mathrm{s}} = \begin{bmatrix} 1 & T_{\mathrm{s}} \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{B}_{\mathrm{s}} = \begin{bmatrix} T_{\mathrm{s}}^2/2 \\ T_{\mathrm{s}} \end{bmatrix}.$$

Remark : Since s, v_{s} and a_{s} are all along the reference paths with one DOF, they are modeled as continuous regardless of the two DOF coordinate systems, e.g., $\{n\}$ or $\{p_j\}$. □

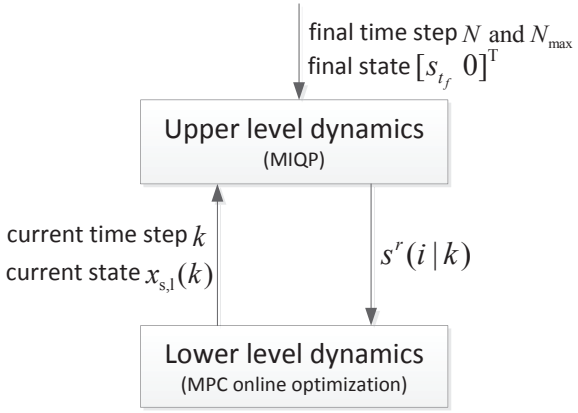


Figure 4.2: Two-level double integrator dynamics.

To distinguish notations in the two levels, we use subscript $\cdot_{\mathrm{s,l}}$ denoting lower level variables and $\cdot_{\mathrm{s,h}}$ indicating higher level variables. In the lower level, given an initial state $\boldsymbol{x}_{\mathrm{s,l}}(k)$, predicted trajectories over a prediction horizon are:

$$\boldsymbol{x}_{\mathrm{s,l}}(i+1|k) = \boldsymbol{A}_{\mathrm{s}}\boldsymbol{x}_{\mathrm{s,l}}(i|k) + \boldsymbol{B}_{\mathrm{s}}u_{\mathrm{s,l}}(i|k), \tag{4.7}$$

for $i = 0,1,...,N_{\mathrm{p}}-1$ with $\boldsymbol{x}_{\mathrm{s,l}}(0|k) = \boldsymbol{x}_{\mathrm{s,l}}(k)$. Prediction model (4.7) is embedded in online MPC optimizations that are to be formulated in Section 4.5.

The higher level shares the same double integrator dynamics (4.6) with the lower level. An MIQP problem is formulated aiming at generating an optimal reference trajectory for the lower level over the next receding prediction horizon, i.e., $s^{\mathrm{r}}(i|k)$, for $i = 1,2,...,N_{\mathrm{p}}$. For the MIQP, we specify main objective as guaranteeing a required arrival at $s_{t_{\mathrm{f}}}$ at time t_{f}. Terminal state $s_{t_{\mathrm{f}}}$ is set as the total length of all the path segments.

Considering limitations of waterborne AGV dynamics, the double integrator dynamics cannot evolve freely either. A waterborne AGV's maximum surge speed $u_{\max}$ is imposed as a state constraint for $v_{\mathrm{s,u}}$ in MIQP. Due to this speed limit, there could be a feasibility issue for a specific ITT task: if the scheduled arrival time is too stringent, the waterborne AGV could not be able to arrive on time even if it sails at its highest speed. In reality, a time window is often assigned to allow for an acceptable delay Δt in terms of a preferable arrival time t_{r}. Finite flexibility is thus set for the arrival time by $t_{\mathrm{f}} \in \left[\underline{t_{\mathrm{f}}}, \overline{t_{\mathrm{f}}}\right]$, where $\underline{t_{\mathrm{f}}} = t_{\mathrm{r}}$ and $\overline{t_{\mathrm{f}}} = t_{\mathrm{r}} + \Delta t$ and we assume by $\overline{t_{\mathrm{f}}}$, the arrival can by all means be achieved. In this case, the problem becomes a constrained optimal control problem with a fixed terminal state and a minimal arrival time [31]. However, the minimal arrival time should be within the time window $\left[\underline{t_{\mathrm{f}}}, \overline{t_{\mathrm{f}}}\right]$. Next, we show how this can be implemented in MIQP using binary variables.

In a discrete time setting, we denote $T_{\mathrm{f}}(k)$ as the calculated arrival time step at time step k, N and $N_{\max}$ corresponding to continuous time $\underline{t_{\mathrm{f}}}$ and $\overline{t_{\mathrm{f}}}$, respectively. Therefore, $T_{\mathrm{f}}(k)$, N and $N_{\max}$ satisfy $T_{\mathrm{f}}(k) \in [N, N_{\max}]$. The cost function regarding the energy and arrival time is separated into two parts:

$$J_{\mathrm{s}}(k) = J_{\mathrm{s}}^1(k) + J_{\mathrm{s}}^2(k), \tag{4.8}$$

where $J_{\mathrm{s}}^1(k)$ is written as a summation from the current time step k to time step $N-1$, i.e.,

$$J_{\mathrm{s}}^1(k) = \sum_{n=k}^{N-1} \left(\|u_{\mathrm{s,h}}(n|k)\|_{w_1}^2 + \|\boldsymbol{x}_{\mathrm{s,h}}(n|k)\|_{\boldsymbol{w}_2}^2 \right) \tag{4.9}$$

subject to, for $n = k, k+1, ..., N-1$

$$\boldsymbol{x}_{\mathrm{s,h}}(n+1|k) = \boldsymbol{A}_{\mathrm{s}} \boldsymbol{x}_{\mathrm{s,h}}(n|k) + \boldsymbol{B}_{\mathrm{s}} u_{\mathrm{s,h}}(n|k), \tag{4.10}$$

$$0 \leq \boldsymbol{x}_{\mathrm{s,h}}(n) \leq u_{\max}. \tag{4.11}$$

Notation $\|\cdot\|_w^2$ stands for weighted vector two norms, e.g., $\|u_{\mathrm{s,h}}(n|k)\|_{w_1}^2 = u_{\mathrm{s,h}}(n|k)^{\mathrm{T}} w_1 u_{\mathrm{s,h}}(n|k)$. Minimization of the two norms of $u_{\mathrm{s,h}}(k)$ and $\boldsymbol{x}_{\mathrm{s,h}}(k)$ aims at optimizing energy efficiency and smoothness of s dynamics. Symbols w_1 and $\boldsymbol{w}_2$ represent the weighting parameter and matrix for $u_{\mathrm{s,h}}(k)$ and $\boldsymbol{x}_{\mathrm{s,h}}(k)$, respectively.

The second part of $J_{\mathrm{s}}(k)$, $J_{\mathrm{s}}^2(k)$, is a summation over $[N, N_{\max}]$, defined as

$$J_{\mathrm{s}}^2(k) = \sum_{n=N}^{N_{\max}} \left(w_3 n \boldsymbol{b}(n-N+1|k) + \|u_{\mathrm{s,h}}(n|k)\|_{w_1}^2 + \|\boldsymbol{x}_{\mathrm{s,h}}(n|k)\|_{\boldsymbol{w}_2}^2 \right), \tag{4.12}$$

where $\boldsymbol{b}(n-N+1|k)$ for $n = N, N+1, ..., N_{\max}$ are binary decision variables satisfying

$$\boldsymbol{b}(n-N+1|k) = \begin{cases} 1, & \text{for } T_{\mathrm{f}}(k) = n, \\ 0, & \text{otherwise.} \end{cases}$$

and

$$\sum_{n=N}^{N_{\max}} \boldsymbol{b}(n-N+1|k) = 1 \tag{4.13}$$

to ensure one arrival time step is selected. This selected arrival time is then the minimal arrival time over $[N, N_{\max}]$. If at time step k, the task is feasible within the preferable arrival time N, then N will be decided as the terminal time of the MIQP. Before the arrival time, $T_{\mathrm{f}}(k)$, the same constraints with $J_{\mathrm{s}}^1(k)$ are imposed to $J_{\mathrm{s}}^2(k)$. Constraints are relaxed after the selected arrival time since the waterborne AGV has stopped. In addition, terminal constraint

$$s_{\mathrm{s,h}}(T_{\mathrm{f}}(k)) = s_{t_{\mathrm{f}}} \tag{4.14}$$

is applied upon $T_{\mathrm{f}}(k)$. We define the above logic as logic constraint C_1 which is modeled for $n_i = N, N+1, ..., N_{\max}$ as:

$$C_1 = \begin{cases} \text{(4.10) and (4.11)}, & \text{for } \sum_{n=N}^{n_i} \boldsymbol{b}(n-N+1|k) = 0, \\ s_{\mathrm{s,h}}(n|k) = s_{t_{\mathrm{f}}}, & \text{for } \sum_{n=N}^{n_i} \boldsymbol{b}(n-N+1|k) = 1. \end{cases} \tag{4.15}$$

A diagram illustrating the timing involved cost of J_{s}^2 is shown in Figure 4.3. Therefore, the final MIQP problem formulated at the higher level is as:

$$\boldsymbol{u}_{\mathrm{s,h}}{}^*(k), \boldsymbol{b}^*(k) = \underset{\boldsymbol{u}_{\mathrm{s,h}}, \boldsymbol{b}}{\operatorname{argmin}} J_{\mathrm{s}}(k), \tag{4.16}$$

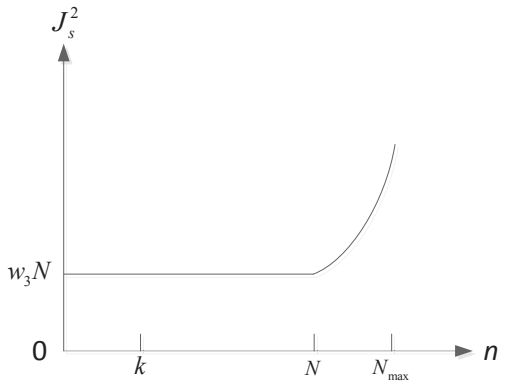


Figure 4.3: Preferable arrival time and a maximum delay.

subject to, for $n = k, k+1, ..., N-1$,

(4.10) and (4.11),

and for $n = N, N+1, ..., N_{\max}$

(4.13) and (4.15).

Remark : As can be observed from the above derivation, the length of the reference generated by the higher level is shortened by one each simulation step. The MPC embedded low level, however, requires an N_{p}-length reference $s^{\mathrm{r}}(k+i|k)$, for $i = 1, 2, ..., N_{\mathrm{p}}$ each step. When the current time is still distant with the scheduled arrival time, the generated reference might remain longer than N_{p}, however, upon arrival, this might cause problems. Therefore, we introduce N_{p} extra time steps in addition to $N_{\max}$, and states $\boldsymbol{x}_{\mathrm{s}}(n)$ over $n = N_{\max}+1, N_{\max}+2, ..., N_{\max}+N_{\mathrm{p}}$ are then constrained to stay as the terminal state, i.e.,

$$s_{\mathrm{s,h}}(n) = s_{t_{\mathrm{f}}}. \tag{4.17}$$

□

4.4 Predictive switching logic

MPC can take into account future situations so that effective actions can be taken at an early stage to avoid undesirable system behaviors. This predictive feature of MPC is useful in our switching reference path segments to avoid overshoots. Based on the path coordinate system models described in Section 4.2, we formulate a predictive switch logic in this section. Before proceeding, two relevant definitions are given first.

Definition 4.1 (Position) *The position of a waterborne AGV is called in path coordinate system $\{p_j\}$ at time step k if $x_{p_j}(k)$, the along-track state in $\{p_j\}$, is not larger than the length of reference path j, i.e.,*

$$x_{p_j}(k) \leq l_j. \tag{4.18}$$

Definition 4.2 (Tracking errors) *For the waterborne AGV to track a geometric reference path j, three kinds of tracking errors are recognized and minimized in online MPC optimizations:*

- *cross-track error $y_{p_j}(k)$ the definition of which has been given in Section 4.2 and $y_{p_j}(k) \to 0$ indicates a convergence to the reference path;*
- *along-track error $s_j(k) - s(k)$ where $s_j(k)$ is the total along-track distance the waterborne AGV has traveled, so its relationship with the along-track state $x_{p_j}(k)$ is:*

$$s_j(k) = x_{p_j}(k) + \left(\sum_{i=1}^{j} l_i - l_j \right); \tag{4.19}$$

- *and heading angle error ψ_{p_j}.*

Gathering the tracking errors, we define

$$\boldsymbol{x}^e_{\mathrm{p}_j}(k) = \left[\begin{array}{ccc} s_j(k) - s_{\mathrm{s,l}}(k) & y_{\mathrm{p}_j}(k) & \psi_{\mathrm{p}_j} \end{array} \right]^{\mathrm{T}}. \tag{4.20}$$

The error vector $\boldsymbol{x}^e_{\mathrm{p}_j}$, which is coordinate system dependent, is minimized in MPC for tracking.

At time step k, considering the waterborne AGV is still in $\{p_j\}$, the initial states $\boldsymbol{x}_{\mathrm{p}_j}(k)$ can be obtained by a transformation of the current measured waterborne AGV states $\boldsymbol{x}(k)$ from $\{n\}$ to $\{p_j\}$ according to (4.4) and (4.5). Future system trajectories $\boldsymbol{x}_{\mathrm{p}_j}(i|k)$ are then predicted in a linear way as (3.13) – (3.14) and (3.18)

Remark : Note that the successive linearization approach in Chapter 3 is applied to path dynamics in this chapter. In particular, since the discretization and linearization theories are not applicable to discontinuous dynamics, all the predicted system trajectories as well as the seed trajectories for linearizations are defined for path dynamics in $\{p_j\}$. Therefore, (3.13) – (3.14) and (3.18) are linearized path dynamics in $\{p_j\}$. Two modifications are as follows:

- Successive linearizations are implemented for the nonlinear path dynamics which uses the path coordinate kinematics (4.1) instead of the inertial coordinate (3.1);
- Initial states for path dynamics are not directly measurable, and are transformed from measured the inertial coordinate states by (4.5).

The successive linearization procedure of path dynamics in this chapter is summarized by Figure 4.4 based on the three steps in Chapter 3. □

If the predicted trajectories to be optimized are indeed all within $\{p_j\}$, e.g., when the waterborne AGV is far away from a switching waypoint, as Figure 4.5a shows, minimization of $\boldsymbol{x}^e_{\mathrm{p}_j}$ in online MPC optimizations realizes reference tracking. However, since MPC looks into the future over a prediction horizon, an initial state close to the switching waypoint would then result in predicted trajectories dispersed in both $\{p_j\}$ and $\{p_{j+1}\}$. In this case, minimizations of $\boldsymbol{x}^e_{\mathrm{p}_j}$ will result in overshoots as Figure 4.5b shows. A transformation of coordinates from $\{p_j\}$ to $\{p_{j+1}\}$ is then necessary. Therefore, based on Definition 4.1, the

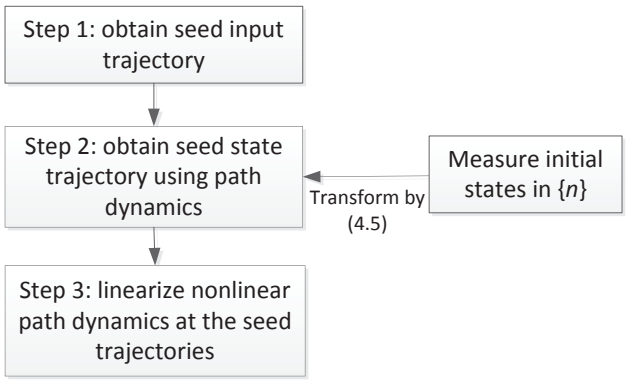


Figure 4.4: Successive linearizations in path coordinates.

following logic is introduced to realize a shift of coordinate system for predicted trajectories and tracking error $\boldsymbol{x}^e_{\mathrm{p}_j}$ so that overshoots are avoided, as Figure 4.5c illustrates.

Define binary decision variable $\boldsymbol{b}_{\mathrm{p}}(k)$ as an $N_{\mathrm{p}} \times 1$ vector at time step k with

$$\boldsymbol{b}_{\mathrm{p}}(i|k) = \begin{cases} 1, & \text{for } x_{\mathrm{p}_j}(i|k) \leq l_j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.21}$$

When the waterborne AGV travels to $\{p_{j+1}\}$, i.e., when $x_{\mathrm{p}_j}(i|k) > l_j$, it is expected to track reference path $j+1$. This logic is expressed as logic constraint C_2 as:

$$C_2 = \begin{cases} \boldsymbol{x}^e_{\mathrm{p}_j}(i|k), & \text{for } \boldsymbol{b}_{\mathrm{p}}(i|k) = 1, \\ \boldsymbol{x}^e_{\mathrm{p}_{j\to j+1}}(i|k), & \text{for } \boldsymbol{b}_{\mathrm{p}}(i|k) = 0. \end{cases} \tag{4.22}$$

where $\boldsymbol{x}^e_{\mathrm{p}_{j\to j+1}}(i|k)$ is the predicted tracking error with respect to reference path $j+1$ while the waterborne AGV is still in $\{p_j\}$, i.e., predicted states $\boldsymbol{x}_{\mathrm{p}_j}(i|k)$ are still derived in $\{p_j\}$. Then according to transformations from $\{p_j\}$ to $\{p_{j+1}\}$ as (4.2) and (4.3),

$$\boldsymbol{x}^e_{\mathrm{p}_{j\to j+1}}(i|k) = \begin{bmatrix} s_{j+1}(i|k) - s_{\mathrm{s},1}(i|k) & y_{\mathrm{p}_{j+1}}(i|k) & \psi_{\mathrm{p}_{j+1}}(i|k) \end{bmatrix}^{\mathrm{T}}, \tag{4.23}$$

where

$$s_{j+1}(k+i|k) = x_{\mathrm{p}_{j+1}}(i|k) + \left(\sum_{jj=1}^{j+1} l_{jj} - l_{j+1} \right), \tag{4.24}$$

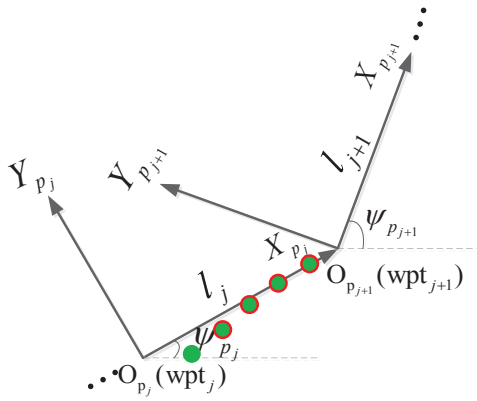
and

$$\begin{bmatrix} x_{\mathrm{p}_{j+1}}(i|k) \\ y_{\mathrm{p}_{j+1}}(i|k) \end{bmatrix} = \begin{bmatrix} \cos(\psi_{j+1} - \psi_j) & \sin(\psi_{j+1} - \psi_j) \\ -\sin(\psi_{j+1} - \psi_j) & \cos(\psi_{j+1} - \psi_j) \end{bmatrix} \begin{bmatrix} x_{\mathrm{p}_j}(i|k) - l_j \\ y_{\mathrm{p}_j}(i|k) - 0 \end{bmatrix}, \tag{4.25}$$

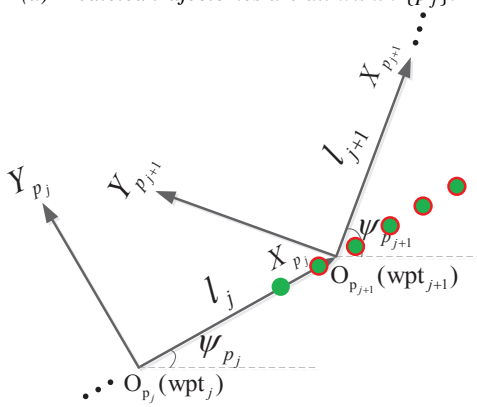
and

$$\psi_{\mathrm{p}_{j+1}}(i|k) = \psi(i|k) - \psi_{j+1}(i|k). \tag{4.26}$$

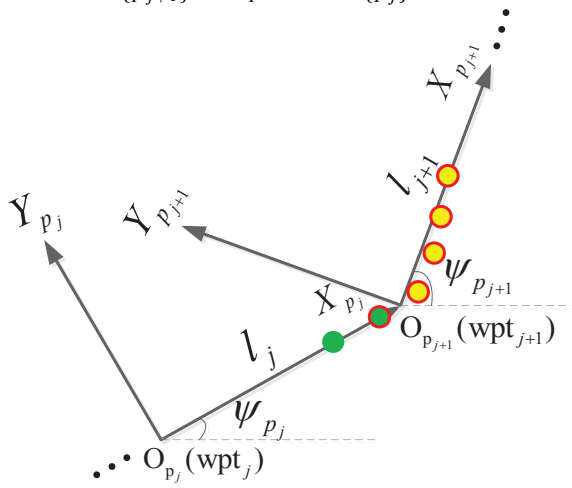
In this way, a solution to the binary variable $\boldsymbol{b}_{\mathrm{p}}(i|k)$ will predictively and optimally determine the waterborne AGV's predicted position in coordinate system $\{p_j\}$ or $\{p_{j+1}\}$. Corresponding tracking errors are then minimized in the online MPC optimizations and overshoots are expected to be avoided as in Figure 4.5c.



(a) Predicted trajectories are all within $\{p_j\}$.



(b) Predicted trajectories are dispersed in $\{p_j\}$ and $\{p_{j+1}\}$ and optimized in $\{p_j\}$.



(c) Predicted trajectories are dispersed and optimized in $\{p_j\}$ and $\{p_{j+1}\}$.

Figure 4.5: Green dot–initial states; red circled dots–states predicted in $\{p_j\}$; red circled green dots– States optimized in $\{p_j\}$; red circled yellow dots – States optimized in $\{p_{j+1}\}$.

4.5 Receding horizon control

This section describes the proposed PPF-ATA algorithm based on MPC. MPC online optimizations compute optimal control inputs based on approximated linearized prediction models. Waterborne AGV behaviors are then updated based on the first element of the optimal control input sequence. This process is repeated until the waterborne AGV arrives at the destination specified by the ITT task. To achieve arrival time awareness and smooth tracking, double integrator dynamics are introduced for path parameterization and generating timing-aware references over the prediction horizon by solving MIQPs. Overshoots are avoided in the proposed MPC framework during switching waypoints by optimizing switching logic related binary decision variables.

To achieve all the control goals presented in Section 4.1, four cost terms are minimized in online MPC optimizations:

1. Path tracking errors as defined in Section 4.4 over the prediction horizon;

2. Too large changes in control inputs which could lead to actuator damages;

3. Kinetic energy consumption which is formulated as $\frac{1}{2}\mathbf{v}^{\mathrm{T}}\boldsymbol{M}\mathbf{v}$ for a surface waterborne AGV, where $\boldsymbol{M}=\boldsymbol{M}_{RB}+\boldsymbol{M}_{A}$ is the mass matrix;

4. Differences between the lower level and higher level double integrator dynamics that might cause delays.

Therefore, for a waterborne AGV in $\{p_j\}$ at time step k, the following MPC optimization problem is solved:

$$\Delta\boldsymbol{u}^{*}(k),\boldsymbol{u}_{\mathrm{s,l}}{}^{*}(k),\boldsymbol{b}_{\mathrm{p}}{}^{*}(k)=\underset{\Delta\boldsymbol{u},\boldsymbol{u}_{\mathrm{s}},\boldsymbol{b}_{\mathrm{p}}}{\operatorname{argmin}}J(k), \tag{4.27}$$

where

$$\begin{aligned}J(k)=\sum_{i=0}^{N_{\mathrm{p}}-1}\Big(&\left\|\boldsymbol{x}_{\mathrm{p}_j}^{e}(i+1|k)\right\|_{\boldsymbol{w_4}}^{2}+\|\Delta\boldsymbol{u}(i|k)\|_{\boldsymbol{w_5}}^{2}+\\&\|\mathbf{v}(i+1|k)\|_{w_6\boldsymbol{M/2}}^{2}+\|\boldsymbol{s}(i+1|k)-\boldsymbol{s}^{\mathrm{r}}(i+1|k)\|_{w7}^{2}\Big),\end{aligned} \tag{4.28}$$

subject to,

$$(3.13), (3.14), (3.18), \tag{4.29a}$$
$$(4.21), (4.22), \tag{4.29b}$$
$$(4.7), \tag{4.29c}$$
$$\Delta \boldsymbol{x}_{\mathrm{p}_j}(k|k) = 0, \tag{4.29d}$$
$$|\boldsymbol{u}(i|k)| \leqslant \boldsymbol{u}_{\max}, \tag{4.29e}$$
$$\boldsymbol{x}_{\mathrm{p}_j,\min} \leqslant \boldsymbol{x}_{\mathrm{p}_j}(i+1|k) \leqslant \boldsymbol{x}_{\mathrm{p}_j,\max}, \tag{4.29f}$$
$$x_{\mathrm{p}_j}(i+1|k) \leqslant \left(\mathrm{obs}_{x_{p_j},\min} - d_{\mathrm{s}}\right) + Mb_{\mathrm{obs},1}, \tag{4.29g}$$
$$-x_{\mathrm{p}_j}(i+1|k) \leqslant -\left(\mathrm{obs}_{x_{p_j},\max} + d_{\mathrm{s}}\right) + Mb_{\mathrm{obs},2}, \tag{4.29h}$$
$$y_{\mathrm{p}_j}(i+1|k) \leqslant \left(\mathrm{obs}_{y_{p_j},\min} - d_{\mathrm{s}}\right) + Mb_{\mathrm{obs},3}, \tag{4.29i}$$
$$-y_{\mathrm{p}_j}(i+1|k) \leqslant -\left(\mathrm{obs}_{y_{p_j},\max} + d_{\mathrm{s}}\right) + Mb_{\mathrm{obs},4}, \tag{4.29j}$$
$$\sum_{n=1}^{4} b_{\mathrm{obs},\,n} \leqslant 3 \text{ and } b_{\mathrm{obs},\,n} \in \{0,1\}. \tag{4.29k}$$

where $\bullet^*(k)$ denote the sequence of optimal variables solved at time step k. In $J(k)$, references for the lower level double integrator dynamics, $\boldsymbol{s}^{\mathrm{r}}(i|k)$ over the prediction horizon are calculated by solving an MIQP problem before solving the online MPC optimization problem. Generally, the time steps involved in the calculated reference $\boldsymbol{s}^{\mathrm{r}}$ are longer than N_{p}. However, only the $\boldsymbol{s}^{\mathrm{r}}(i+1|k)$ is necessarily fed to $J(k)$. Constraints (4.29a) are equality constraints of the approximated linearized prediction models of nonlinear path coordinate dynamics. Constraints (4.29b) are the logic constraints for formulations of tracking errors in different reference path frames, as derived in Section 4.4; initial incremental state $\Delta \boldsymbol{x}_{\mathrm{p}_j}(0|k)$ is set to zero as (4.29d) because $\boldsymbol{x}_{\mathrm{p}_j}(0|k) = \boldsymbol{x}^0_{\mathrm{p}_j}(0|k)$ and both of them are equal to $\boldsymbol{x}_{\mathrm{p}_j}(k)$ which is the current "measured"[2] state; system limitations on control inputs and states due to system physical limits on maximum actuator forces/moment and maximum speed, etc, are imposed by (4.29e) and (4.29f), respectively; obstacle avoidance for static obstacles which causes delays to a schedule are formulated as (4.29g) – (4.29k) where $\left(\mathrm{obs}_{x_{\mathrm{p}_j},\min}, \mathrm{obs}_{y_{\mathrm{p}_j},\min}\right)$ and $\left(\mathrm{obs}_{x_{\mathrm{p}_j},\max}, \mathrm{obs}_{y_{\mathrm{p}_j},\max}\right)$ are the coordinates in $\{p_j\}$ for the left-low and right-up corner of a rectangular obstacle, respectively; $b_{\mathrm{obs},\,n}$ and M are binary variables and a big value, respectively for an convex obstacle avoidance formulation [117]. Since the avoidance constraints are only applied at discrete time steps, a safety margin $d_{\mathrm{s}} = u_{\max} T_{\mathrm{s}} / 2\sqrt{2}$ is implemented to avoid crossings in corners [53]. Note here that obstacle avoidance constraints are imposed to the center of the waterborne AGV without considering specific waterborne AGV shapes. However, it is assumed that waterborne AGV sizes have been taken into account when obstacle areas are defined. Therefore, as long as the trajectory of waterborne AGV's center is outside obstacle areas, the waterborne AGV is safe.

At each time step k, two MIQPs need to be solved: one is the upper level timing-aware reference generation problem (4.16) – (4.15) and the other is the online MPC optimiza-

[2] $\boldsymbol{x}_{\mathrm{p}_j}(k)$ is not directly measurable but transformed from $\boldsymbol{x}(k)$ by (4.4) and (4.5)

Algorithm 4.1 Predictive path following with arrival time awareness (PPF-ATA)

Initialization at path coordinate system $j = 1$ at time step $k = 0$;
Solve problem (4.16) – (4.15) to obtain $\boldsymbol{x}_{\mathrm{s}}^{r}(i)$ for $i = 1, 2, ..., N_{\mathrm{p}}$;
while $\boldsymbol{x}(k) \neq \boldsymbol{x}(t_{\mathrm{f}})$ **do**
  while $x_{\mathrm{p}_j}(k) \leq l_j$ **do**
    Measure and transform current states $\boldsymbol{x}(k)$ to $\boldsymbol{x}_{\mathrm{p}_j}(k)$ in $\{p_j\}$;
    Obtain linearized prediction models as Figure 4.4;
    Solve optimization problem (4.27)-(4.29) to determine $\boldsymbol{u}^*(k)$, ${\boldsymbol{u}_{\mathrm{s}}}^*(k)$;
    Apply the first element $\boldsymbol{u}^*(k|k)$ to waterborne AGV dynamics (3.1) and (3.5);
    $k = k + 1$;
  end while
  $j = j + 1$;
end while

tion problem (4.27)-(4.29). With reasonable problem size for one waterborne AGV, the two MIQPs can be solved efficiently by standard solvers. Each time a new optimization problem is formulated given the current new measurements; a sequence of optimal control inputs $\boldsymbol{u}^*(k) = \Delta\boldsymbol{u}^*(k) + \boldsymbol{u}^0(k)$ is calculated which drives predicted system outputs close to set references to achieve design requirements. The first element of this optimal control sequence, i.e., $\boldsymbol{u}^*(k|k)$ is applied to the real system (3.1) and (3.5). Time is then shifted one step forward and the above procedures are repeated at the new time step to formulate a receding horizon law. Convergence to the reference path and timing aware of arrival at the destination in an economical way is thus guaranteed. System constraints are also well considered in online MPC optimizations. The overall algorithm for the problem of PPF-ATA is summarized in Algorithm 4.1.

The condition in the outer *while* loop $\boldsymbol{x}(k) \neq \boldsymbol{x}(t_{\mathrm{f}})$ means that a waterborne AGV has not arrived at the final destination yet and

$$\boldsymbol{x}(t_{\mathrm{f}}) = \begin{bmatrix} \boldsymbol{\eta}_{t_{\mathrm{f}}}{}^{\mathrm{T}} & \boldsymbol{\nu}_{t_{\mathrm{f}}}{}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{4.30}$$

where $\boldsymbol{\eta}_{t_{\mathrm{f}}}$ is the final pose dependent on reference path information and $\boldsymbol{\nu}_{t_{\mathrm{f}}} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$. The PPF-ATA controller based on Algorithm 4.1 designed for waterborne AGVs is shown in Figure 4.6. Note that the original nonlinear waterborne AGV dynamics in $\{n\}$ are used in the closed-loop simulation.

4.6 Simulation results and discussion

In this section, we present simulation results of two typical ITT scenarios to illustrate how the PPF-ATA controller works and to demonstrate its potential for ITT. For the first simulation, the controller is given a feasible ITT task, which means the scheduled preferable arrival time can be achieved by the waterborne AGV. In the second simulation, an infeasible ITT task is set where the preferable arrival time cannot be met even if the waterborne AGV sails at the maximum speed all the time without any obstacles. In the latter case, we show how the waterborne AGV achieves the task with a minimum delay regarding the preferable arrival time.

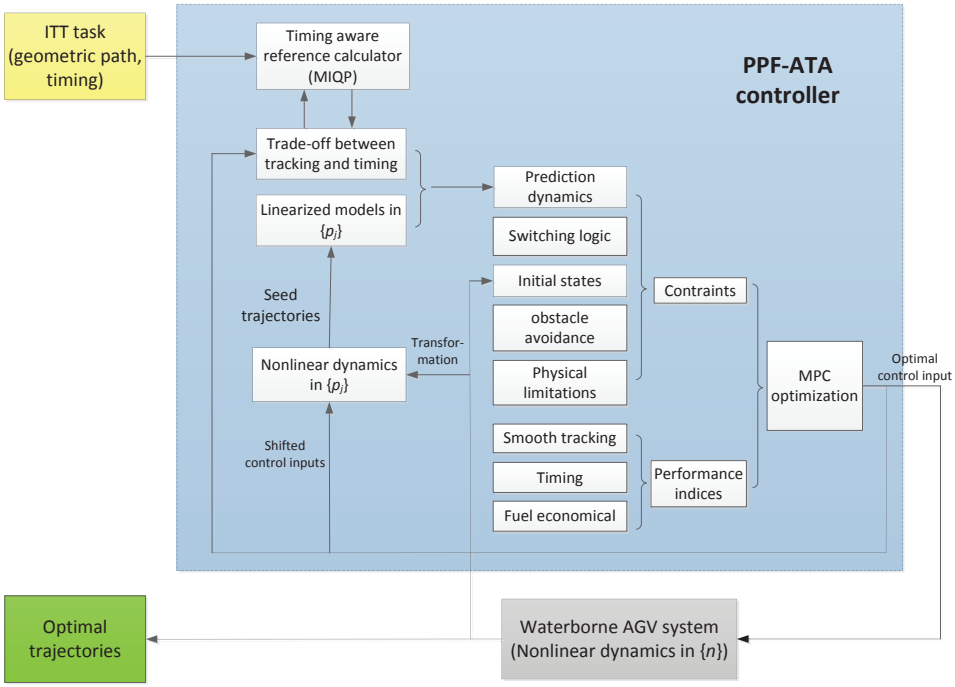


Figure 4.6: PPF-ATA controller for waterborne AGVs.

For both simulations, we set an ITT task from APM terminal to Euromax Terminal in the port of Rotterdam, as shown in Figure 4.7. Distributing ITT over waterborne AGVs is of practical interest in this scenario since these two terminals are not connected by land, and even if they would be connected, the distance by land is much longer than by water. The reference path consists of several straight-line segments. Simulations are implemented based on a $1:70$ small scaled marine surface vehicle model, CSII [121] since all of the necessary parameters for models in (3.1) and (3.5) have been experimentally identified. Simulation results based on CSII are then scaled-up according to Froude scaling law [80], e.g., $1:70$ for length (m) and $1:\sqrt{70}$ for time (s), for the real scale quantities. The reference path information[3] for both of the two ITT tasks is then given as Table 4.1.

Reference path details including l_j and ψ_j can then be calculated from the waypoints given in Table 4.1. Considering that in a real situation, the waterborne AGV will not stop with a heading angle decided by Intermediate waypoint 2 and Euromax Terminal, but a heading angle required by berthing at the terminal, see the red circle in Figure 4.7. Intermediate waypoint 3 is therefore introduced to produce the final reference heading angle. Therefore, final pose in (4.30) is given as $\boldsymbol{\eta}_{t_f} = \begin{bmatrix} -334.7 & 1786.5 & 3.04 \end{bmatrix}^{\mathrm{T}}$. We show that this berthing behavior can also be well achieved by our PPF-ATA controller. In addition, two static obstacles are placed along the path, which causes unexpected delays. One of them is placed half way of the first line segment, and the other half way of the third line

[3]The positions in latitude/longitude are obtained from Google Earth and then converted to inertial frame coordinates with APM Terminal as the origin.

Figure 4.7: APM terminal and Euromax Terminal at Maasvlakte 2 in the port of Rotterdam from Google Earth [35].

Table 4.1: ITT scenario for a singe waterborne AGV.

	Lat./Lon.	(x_n, y_n) (m)
APM Terminal	(51.9578°, 4.0417°)	(0, 0)
Waypoint 1	(51.9614°, 4.0533°)	(798.0, 404.7)
Waypoint 2	(51.9655°, 4.0538°)	(829.1, 852.1)
Waypoint 3	(51.9734°, 4.0390°)	(-187.5, 1731.5)
Euromax Terminal	(51.9739°, 4.0368°)	(-334.7, 1786.5)

segment.

The experiments also share the same MPC controller settings with a prediction horizon $N_\mathrm{p} = 20$. Weight parameters are given as:

$$w_1 = 1,\ \boldsymbol{w}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},\ w_3 = 1000,$$
$$\boldsymbol{w}_4 = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 100 \end{bmatrix},\ \boldsymbol{w}_5 = \boldsymbol{I}_{3\times 3},\ \boldsymbol{w}_6 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ w_7 = 100. \quad (4.31)$$

The waterborne AGV is initially positioned at $(70, 0)$ with $\psi = \pi$ and zero velocity, i.e., $\boldsymbol{x}_0 = \begin{bmatrix} 70 & 0 & \pi & 0 & 0 & 0 \end{bmatrix}^\mathrm{T}$. System sampling time $T_\mathrm{s} = 1$s. System constraints are set as:

$$\begin{bmatrix} 0 \\ -0.84 \\ -15\pi/180 \end{bmatrix} \leqslant \boldsymbol{v} \leqslant \begin{bmatrix} 1.67 \\ 0.84 \\ 15\pi/180 \end{bmatrix},\ \text{and } |\boldsymbol{\tau}_\mathrm{max}| = \begin{bmatrix} 686000 & 686000 & 36015000 \end{bmatrix}^\mathrm{T}.$$

Algorithms in this chapter are implemented using YALMIP (version 20131002) [66] in MATLAB 2011b [75]. Optimization problems are solved by Gurobi (version 5.6 academic) [37]. All the simulations are run on a platform with Intel (R) Core (TM) i5-3470 CPU @3.20 GHz.

4.6.1 Simulation experiment 1: Feasible ITT task

The total reference path length is 2844.6 m and the preferable arrival time at destination is 2510 s after departure from the origin. Therefore, if no unexpected events happen, an average speed of 1.13 m/s should be attained, which is within the maximum speed range of the waterborne AGV system, and thus is deemed as a feasible ITT task. However, considering the waterborne AGV cannot sail at this average speed all the time and delays might still happen due to unexpected events, the higher level MIQP problem is solved based on an acceptable delay tolerance of 167 s. However, delays with respect to 2510 s are penalized in the MIQP problem. The first simulation is run based on this task to achieve smooth path tracking with arrival time awareness in an economical way.

Path tracking performance

Path tracking performance of the PPF-ATA controller is illustrated by smooth convergence to reference paths when there are off-sets and small deviations when the waterborne AGV is on track. Besides, overshoots during switching of reference line segments are well avoided, which also demonstrates the controller's capability of path following.

Figure 4.8 shows how the waterborne AGV accurately tracks the reference path. Yellow heptagons represent the controlled waterborne AGV that are plotted according to waterborne AGV poses at certain time steps. Figure 4.9 further illustrates the path reference tracking performance by showing cross-track and path heading angle errors which are the second and third term in the error vector (4.20), respectively. Small cross-track errors around zero are observed in general with an average value of 1.76 m. Large errors are observed for both

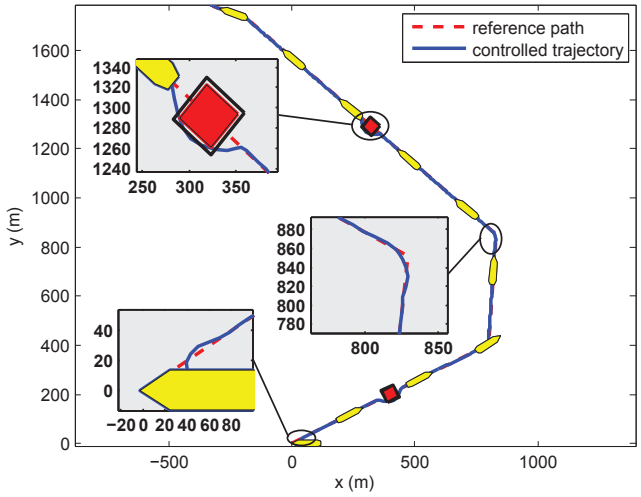


Figure 4.8: Tracking performance of task 1.

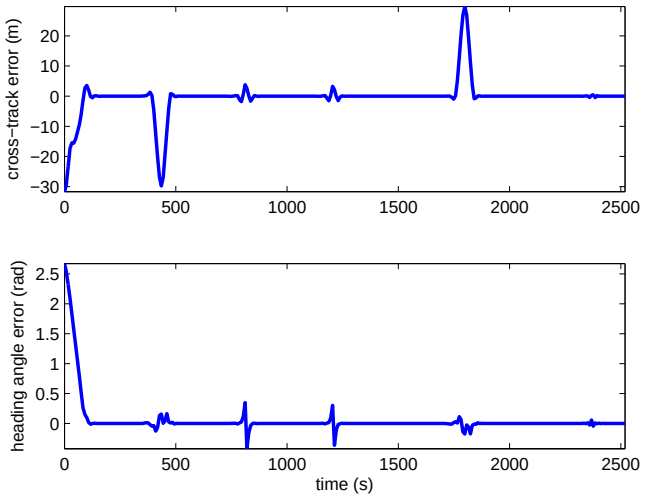


Figure 4.9: Cross-track and heading angle errors.

cross-track and heading at the beginning and also around obstacle areas because there is an initial offset and obstacle avoidance are implemented as hard constraints to guarantee safety. Other relatively smaller deviations in Figure 4.9 are due to switches at the three intermediate waypoints.

The three boxes in Figure 4.8 along the path are zoom-ins of waterborne AGV behaviors at starting point, switching at intermediate waypoint 2, and around the second obstacle, respectively. In boxes 1 and 3, it can be observed that the waterborne AGV is able to

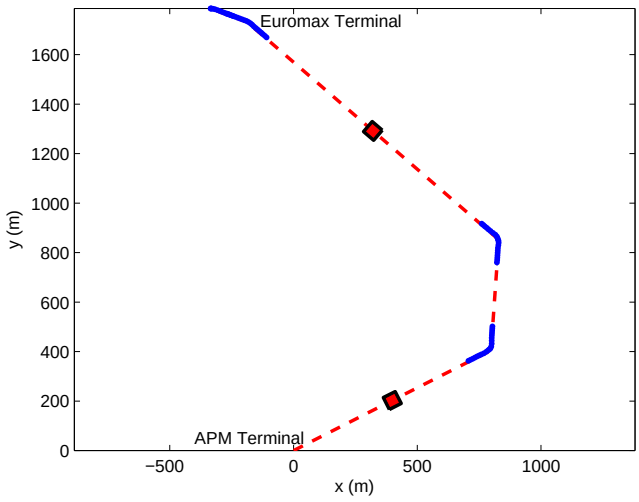


Figure 4.10: Predicted waterborne AGV trajectory over the prediction horizon at one of the time steps during switching.

converge to the reference path smoothly with an initial offset or after a necessary offset to avoid obstacles. This is because the lower level double integrator dynamics always "slow down" to "wait for" the waterborne AGV if the waterborne AGV is in a situation with low speed, e.g., at the starting point, avoiding an obstacle.

During switching of the reference line segments, as shown in the second box, the controlled waterborne AGV trajectory can also match the reference path well with negligible deviations and almost no overshoots. This is due to MPC's predictive feature. In Figure 4.10, predicted waterborne AGV trajectories over the prediction horizon at one time step close to switching waypoints are plotted. If there are no model mismatches between the real system model and the model used for prediction, and if there are no external disturbances, the future system trajectories will be exactly like the one predicted at the current time step, which means that the real waterborne AGV trajectory will also switch successfully. Although successively linearized prediction models inevitably result in model mismatches, the successive linearization framework by conducting the linearization of the nonlinear dynamic system about a shifted optimal trajectory has significantly reduced linearization errors. Therefore, real waterborne AGV trajectories are also expected to have successful switches as in Figure 4.10. Box 2 in Figure 4.8 confirms this.

Arrival time awareness

The "slow down" of lower level double integrator dynamics for smooth path tracking causes delays. However, the delays can be compensated after the waterborne AGV is not so "lagging-behind" by minimizing the error between lower and higher level double integrator dynamics. The higher level considers current new distance-to-go and time-to-go such that timing aware references are generated. Figure 4.11 illustrates this. At the starting time

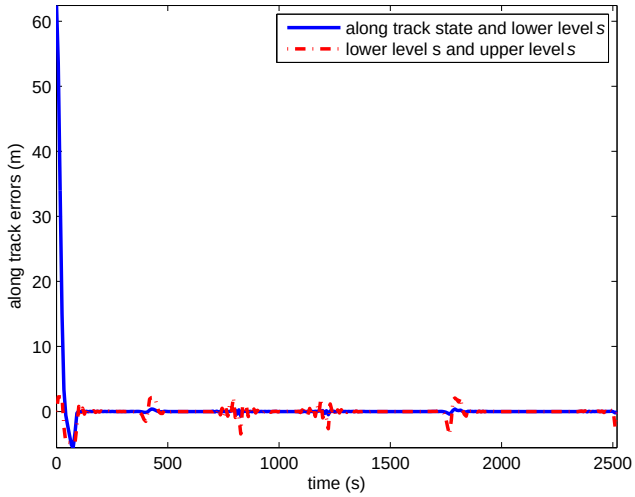


Figure 4.11: Along-track errors.

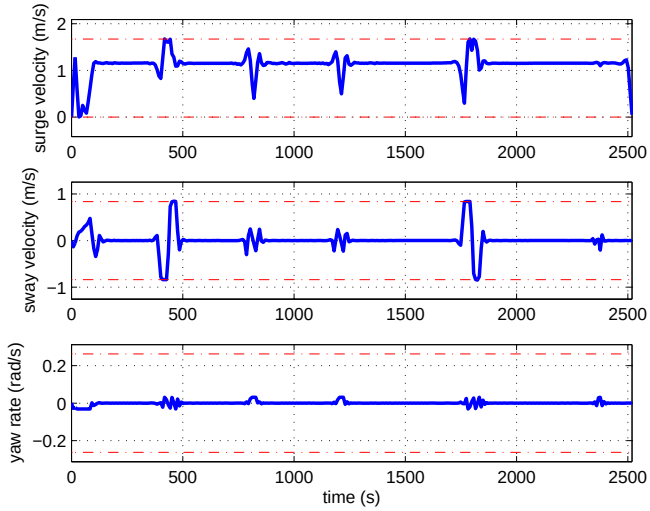


Figure 4.12: Waterborne AGV surge, sway velocities, and yaw rate.

and when the first and second obstacle avoidance happen, both along-track errors and lower-/higher level tracking errors see some fluctuations, but both of them return to an approximate zero afterwards. Moreover, the waterborne AGV arrives at the destination at $t = 2518$ s with eight second's delay which is 0.33% of the total time.

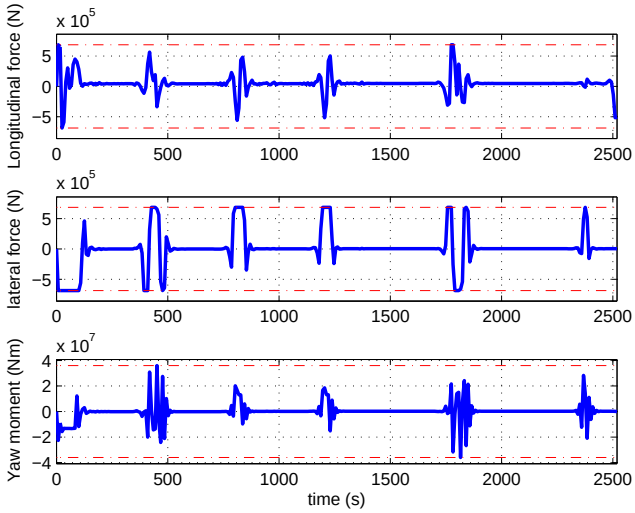


Figure 4.13: Waterborne AGV surge, sway forces, and yaw moment.

Energy consumption and system constraints

The objectives of good path following performance and arrival time awareness are achieved in an energy efficient way within system limits. In Figure 4.12, system velocities all maintain almost constant except for fluctuations at initial, obstacle and reference switching points. Since all the MPC optimization problems are successfully solved, the velocities are optimal values in the feasible region defined by system constraints. Comparisons on energy consumption of the two experiments are presented in Section 4.6.2. System physical constraints are also well satisfied in our scheme. Actuator inputs are shown in Figure 4.13. Same as in Figure 4.12, all the parameters are within the system limitations.

4.6.2 Simulation experiment 2: Infeasible ITT task

In this simulation, the waterborne AGV also needs to follow the scaled reference path from APM Terminal to Euromax Terminal. However, the scheduled preferable arrival time is set to be only 1673 s after departure. Therefore, even if no unexpected events happen and the waterborne AGV is right on the path with a heading angle tangent to the path, the waterborne AGV still needs to sail at an average speed of 1.7 m/s all the time. However, the waterborne AGV has a maximum surge speed of 1.67 m/s let alone the effects of current, off-track positions and zero velocities as initial states, and unexpected events such as obstacle avoidance. Therefore, this ITT task is defined as infeasible. Similarly with Experiment 1, however, we append 167 s as an acceptable maximum arrival time, which results in an average speed of 1.55 m/s. Simulation results below illustrate how the PPF-ATA controller also works well in this scenario to achieve smooth path tracking, arrival time awareness, and energy efficiency.

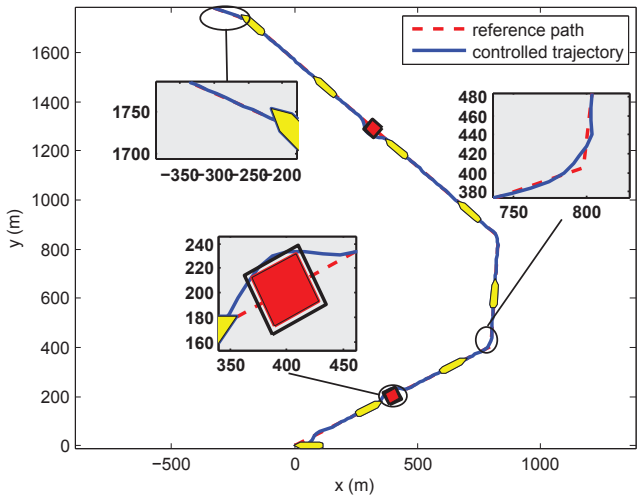


Figure 4.14: Tracking performances.

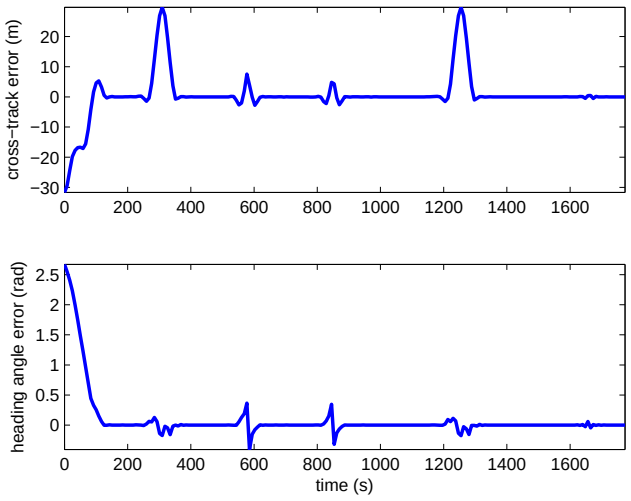


Figure 4.15: Cross-track and heading angle errors.

Path tracking performance

Similar with Experiment 1, accurately tracking of the reference path is observed in this case as Figure 4.14 shows. Three boxes in this figure along the path are zoom-ins of waterborne AGV behaviors around the first obstacle, switching at intermediate waypoint 1, and during the final destination area, respectively. Again, smooth tracking and convergence to the reference path are achieved including areas around the starting point, obstacle and during switches. Figure 4.15 further illustrates the path convergence performance by showing

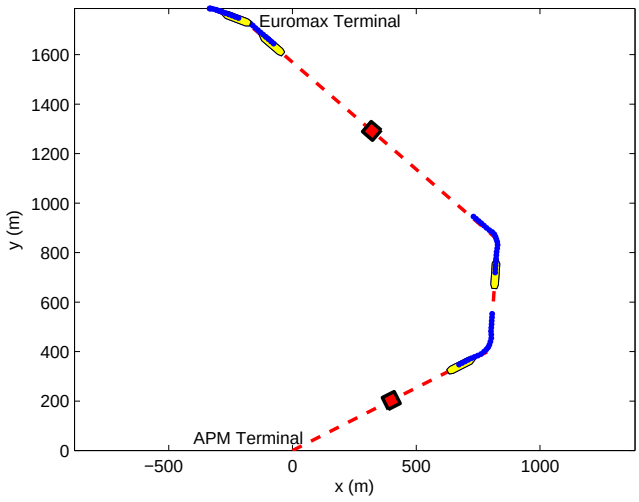


Figure 4.16: Predicted waterborne AGV trajectory over the prediction horizon at one of the time steps during switching.

cross-track and heading angle errors along time. In this experiment, cross-track errors are with an average value of 2.75 m. Relatively obvious deviations in both sub-figures of Figure 4.15 are due to the initial offset, obstacles, and switches at the three intermediate waypoints.

The second box demonstrates the switching of reference paths has been successful with almost no overshoots. Compared with the switch box in Figure 4.8, larger deviations are observed, which can also be observed by comparing the errors caused by switches in Figures 4.9 and 4.15. This is because when the arrival time is set shorter, waterborne AGV needs to sail at a higher speed, which leads to larger errors. Predicted waterborne AGV trajectories over the prediction horizon at one time step near switching waypoints for this ITT task are shown as Figure 4.16.

In the third box, waterborne AGV trajectories can also well follow the last line segment which has been added for a berthing behavior. Figure 4.17 further shows the heading angle trajectories which illustrate that the waterborne AGV stops at the destination terminal with a berthing angle.

Arrival time awareness

Figure 4.18 shows along-track errors in this task. Again, both along-track errors and lower-/higher level tracking errors see some fluctuations during starting, obstacle, and switching areas, but both of them return to an approximate 0 afterwards. Compared to Figure 4.11, the times when fluctuations happen due to obstacles and switches are earlier. This is because in Experiment 2, the waterborne AGV is sailing at a higher speed.

However, in this experiment, the waterborne AGV is still able to meet the timing requirement and arrives at the destination at $t = 1774$ s with a 6% delay with respect to the expected arrival time. Figure 7.10 shows how the arrival times calculated by higher level

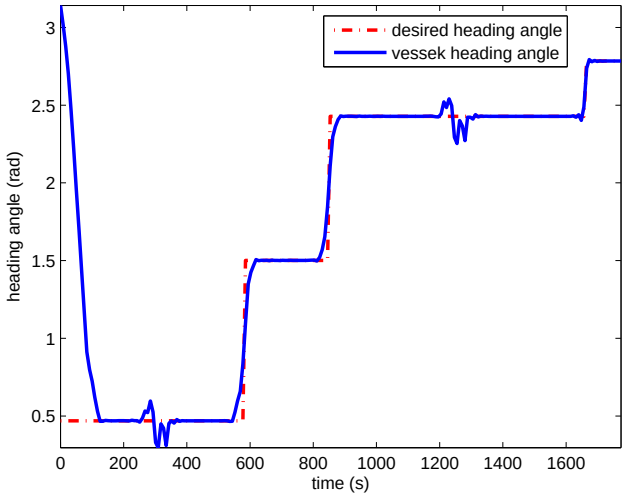


Figure 4.17: Waterborne AGV heading angles.

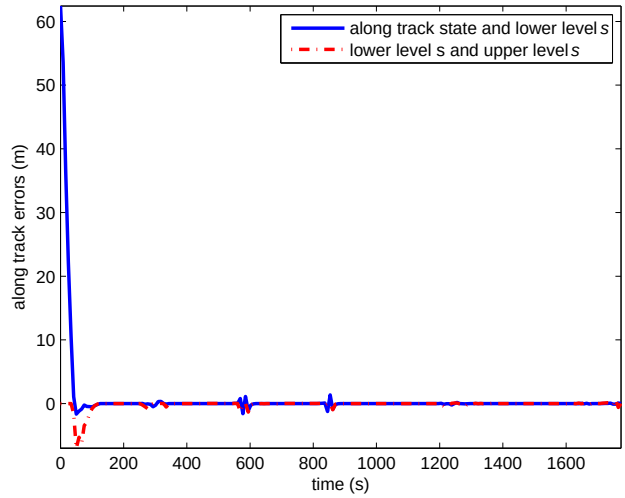


Figure 4.18: Along-track errors.

MIQP changes every time an delay event, e.g., obstacles and switches, happens.

Energy consumption and system constraints

The total kinetic energy consumption calculated according to Section 4.5 for task 1 and task 2 is 1.8×10^6 kJ and 2.6×10^6 kJ, respectively. Again, since we are solving repetitive constrained optimization problems which are all successfully solved, it is sufficient to conclude

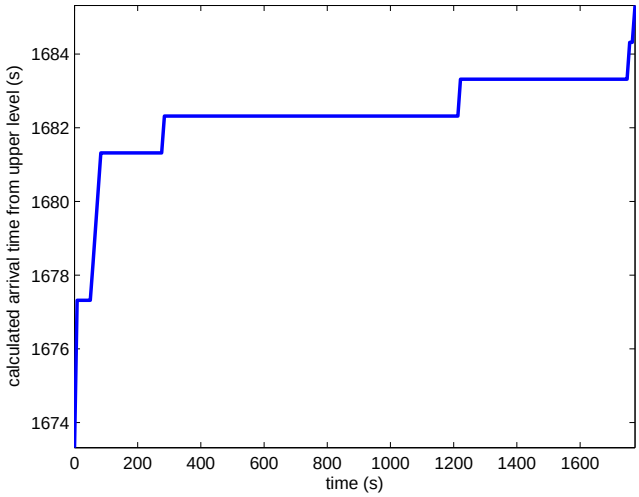


Figure 4.19: Calculated arrival time from the higher level.

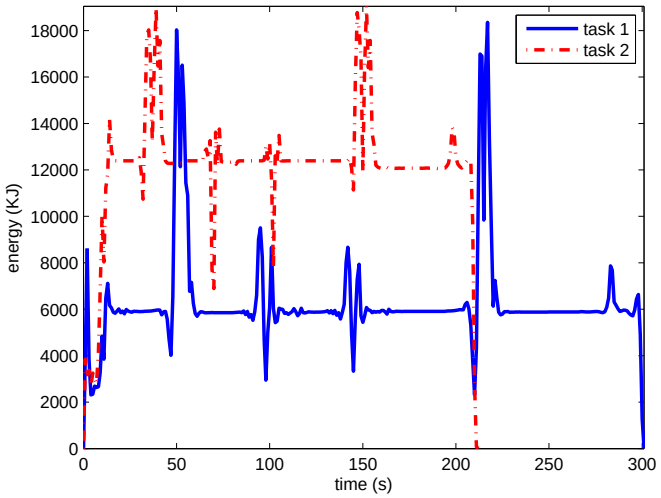


Figure 4.20: Comparison of consumed energy.

that the energy consumption is optimal in a sense that they are the smallest within the system constraints. Although for the second ITT task, the cumulative time is shorter, it still has a much larger total energy consumption. Comparisons of the time-wise energy consumption of them are presented as Figure 4.20. It is clear that the waterborne AGV in the second task is consuming more energy all the time so that it can fulfill the ITT task on time.

Velocity and actuator force trajectories for this experiment are shown as Figure 4.21 and

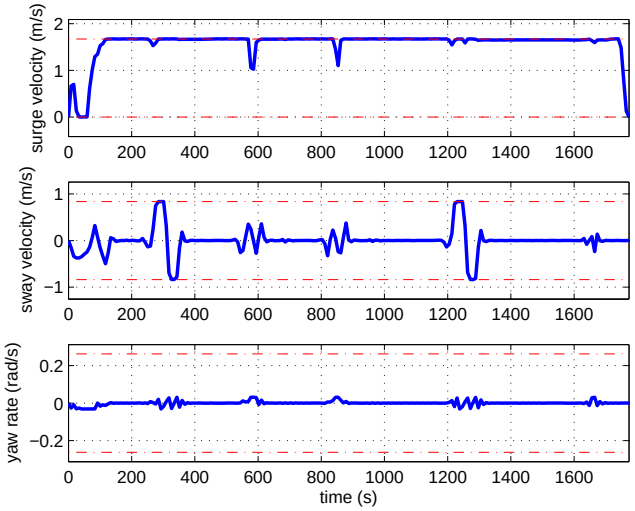


Figure 4.21: Waterborne AGV surge, sway velocities and yaw rate.

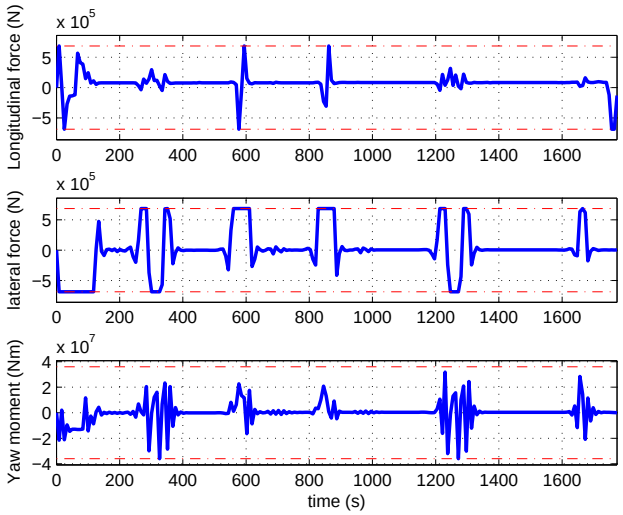


Figure 4.22: Waterborne AGV surge, sway forces and yaw moment.

Figure 4.22, respectively. Again, all the parameters are within the system limits.

4.7 Conclusions

A predictive path following with arrival time awareness (PPF-ATA) controller for a single waterborne AGVs carrying out an assigned ITT task has been proposed in this chapter. The

proposed approach answers the second Key Research Question in Chapter 1 by considering smooth path tracking, arrival time awareness, and energy efficiency as the performance criteria. In our two simulation experiments based on ITT scenarios in the port of Rotterdam, these conflicting objectives have been achieved in a systematic way by the proposed PPF-ATA controller. For both experiments, smooth path tracking behaviors are observed with average cross-track errors 1.76 m and 2.75 m, respectively. The waterborne AGV in the two simulations has 0.33% and 6% delays with respect to the expected arrival times, respectively. The relatively large delay in simulation 2 is due to the physical speed limits and time consuming obstacle avoidance. Moreover, overshoots are avoided during switching reference paths by taking advantage of the predictive feature of MPC and an along-track state involved switching logic. The method proposed in this chapter provides a comprehensive solution applicable to problems on path following with timing requirements including but not limited to waterborne AGVs for ITT.

The PPF-ATA controller proposed for a single waterborne AGV in this chapter will be extended to multiple waterborne AGVs in Chapter 5 and to waterborne AGVs with uncertainties in Chapter 6.

Chapter 5

Cooperative distributed waterborne AGVs

This chapter extends the scenario of a single waterborne AGV considered in Chapter 4 to multiple waterborne AGVs carrying out Inter Terminal Transport (ITT) tasks. The network of multiple waterborne AGVs is modeled as in Chapter 3. The control goal is that waterborne AGVs minimize an overall objective in a cooperative distributed way. Simulation results are presented to illustrate the effectiveness of the proposed algorithms.

The research discussed in this chapter is based on [148, 149].

5.1 Introduction

A single waterborne AGV is controlled by the proposed predictive path following with arrival time awareness (PPF-ATA) controller to fulfill an assigned ITT task in Chapter 4. Control goals such as smooth path tracking, timing, and energy efficiency are well achieved for the waterborne AGV when maneuvering independently. However, as discussed in Chapter 2, usually a fleet of waterborne AGVs will be deployed for carrying out multiple ITT tasks. When waterborne AGVs are in the proximity of each other, couplings arise, as modeled in Chapter 3, which impede independent decision making. System-wide control can be approached in centralized, decentralized, and distributed ways [89]. The advantages of distributed control for waterborne AGVs have been discussed in Chapter 2. The main challenge lies in the satisfaction of collision avoidance couplings in a distributed and preferably parallel way for equally treated waterborne AGVs.

In this chapter, based on the PPF-ATA controller proposed for one single waterborne AGV in Chapter 4, a distributed PPF-ATA controller is proposed for cooperative waterborne AGVs carrying out multiple ITT tasks. ITT using waterborne AGVs is modeled by time-varying graphs, see Chapter 3 and Figure 3.2. Cooperative distributed computations based on these graphs are realized in the framework of distributed MPC where parallelism is achieved following the Alternating Direction Method of Multipliers (ADMM) [12]. A fast ADMM approach iteratively approximating global information in local problems is proposed to obtain better convergence rates than that of the conventional ADMM. Simulation

results for an ITT case study illustrate the effectiveness of the proposed algorithms for distributed MPC of time varying networks in general and cooperative distributed waterborne AGVs in particular.

The remainder of this chapter is organized as follows. A centralized formulation for cooperative waterborne AGVs based on the PPF-ATA controller in Chapter 4 is first presented in Section 5.2. Distributed solutions based on ADMM and fast ADMM are then proposed in Section 5.3 and Section 5.4, respectively. In Section 5.5, simulation experiments and results are discussed, followed by concluding remarks in Section 5.6.

5.2 A centralized formulation

When multiple waterborne AGVs are scheduled for carrying out multiple ITT tasks, besides the control goals of a single waterborne AGV, i.e., smooth path tracking, arrival time awareness, and energy efficiency, extra requirements are to be satisfied:

- a safety distance between waterborne AGVs;
- minimal overall energy consumption; and
- distributed parallel computations.

A centralized solution that satisfies the first two requirements can be formulated based on the models introduced in Chapter 3, and the proposed PPF-ATA controller in Chapter 4 as:

$$\min \sum_{s=1}^{n_{\mathcal{G}}(k)} J_{\mathcal{G}_s}\left(\boldsymbol{X}_{\mathcal{G}_s}(k), \boldsymbol{U}_{\mathcal{G}_s}(k)\right), \tag{5.1}$$

where

$$\begin{aligned} J_{\mathcal{G}_s}\left(\boldsymbol{X}_{\mathcal{G}_s}(k), \boldsymbol{U}_{\mathcal{G}_s}(k)\right) &= \sum_{p\in\mathcal{V}_s(k)} J_p\left(\boldsymbol{x}_p(k), \Delta\boldsymbol{u}_p(i|k)\right) \\ &= \sum_{p\in\mathcal{V}_s(k)} \left\|\boldsymbol{x}_p^e(i|k)\right\|_{\boldsymbol{w_1}}^2 + \left\|\Delta\boldsymbol{u}_p(i|k)\right\|_{\boldsymbol{w_2}}^2 + \left\|\boldsymbol{x}_p^{\mathbf{v}}(i|k)\right\|_{w_3}^2 \end{aligned} \tag{5.2}$$

subject to for $s = 1, 2, \cdots, n_{\mathcal{G}}(k)$

$$\Delta\boldsymbol{x}_p(i|k) \in \mathcal{C}_{x_p}(k) \qquad \forall p \in \mathcal{V}(k), \tag{5.3}$$

$$\Delta\boldsymbol{u}_p(i|k) \in \mathcal{C}_{u_p}(k) \qquad \forall p \in \mathcal{V}(k), \tag{5.4}$$

$$(\Delta\boldsymbol{r}_p(i|k), \Delta\boldsymbol{r}_q(i|k)) \in \mathcal{C}_{r_{p,q}}(k) \qquad \forall e_{p,q}(k) = 1, e_{p,q}(k) \in \mathcal{E}_s(k), \tag{5.5}$$

where local physical limitations due to (3.3) on perturbation states and control inputs are represented by convex sets $\mathcal{C}_{x_p}(k)$ and $\mathcal{C}_{u_p}(k)$ as (5.3) and (5.4), respectively. Similarly, collision avoidance constraints (3.19) on perturbation position variables are represented by convex sets $\mathcal{C}_{r_{p,q}}(k)$ as (5.5).

The total cost in (5.1) is a summation of costs over all subgraphs and the subgraph cost (5.2) is a summation of local costs over all waterborne AGVs in the current subgraph. Local cost functions are convex minimizing path following errors, control input changes, and

kinetic energy consumption. For ease of notation, states and control inputs for $p \in \mathcal{V}_s(k)$ are represented compactly by $\boldsymbol{X}_{\mathcal{G}_s}(k)$ and $\boldsymbol{U}_{\mathcal{G}_s}(k)$, respectively. Centralized problem 5.1 – 5.5 is coupled due to pairwise collision avoidance constraints (5.5). Distributed solutions are proposed next to decompose this centralized problem to satisfy the third requirement.

5.3 Cooperative distributed waterborne AGVs

Centralized problem (5.1) – (5.5) can be decomposed into $n_{\mathcal{G}}(k)$ subgraph problems (5.2) – (5.5) that are independent of one another. Subgraphs that are not singleton, i.e., $n_s(k) > 1$, have coupling collision avoidance constraints (5.5) which prohibit a further distributed solution. This section proposes a distributed PPF-ATA control approach based on the iterative decomposition-coordination procedure of ADMM for the problem over subgraphs.

5.3.1 Derivation of ADMM for waterborne AGVs

For each subgraph $\mathcal{G}_s(k) = (\mathcal{V}_s(k), \mathcal{E}_s(k))$, we assign one of the waterborne AGVs $p \in \mathcal{V}_s(k)$ as the coordinator to take care of the couplings. All waterborne AGVs are able to carry out computations simultaneously and communicate with the coordinator. The coordinator broadcasts solutions towards which local solutions are regularized by adjusting a Lagrange multiplier and an augmented quadratic penalty term until consensus is achieved.

For each waterborne AGV $p \in \mathcal{V}_s(k)$, we introduce a copy of the perturbation position variables as $\Delta\hat{\boldsymbol{r}}_p(i|k)$. Then centralized problem (5.2) – (5.5) is equal to the same problem subject to an additional constraint

$$\Delta\hat{\boldsymbol{r}}_p(i|k) = \Delta\boldsymbol{r}_p(i|k). \tag{5.6}$$

The augmented Lagrangian [12] that relaxes (5.6) is:

$$\begin{aligned}\mathcal{L}_\rho(k) = \sum_{p\in\mathcal{V}_s(k)} \Big(& J_p\left(\boldsymbol{x}_p(k), \Delta\boldsymbol{u}_p(i|k)\right) + \boldsymbol{\lambda}_{p,r}(i|k)^{\mathrm{T}} \left(\Delta\boldsymbol{r}_p(i|k) - \Delta\hat{\boldsymbol{r}}_p(i|k)\right) \\ & + \rho/2 \left\|\Delta\boldsymbol{r}_p(i|k) - \Delta\hat{\boldsymbol{r}}_p(i|k)\right\|_2^2 \Big) + I_{\mathcal{C}}(k),\end{aligned} \tag{5.7}$$

where $\boldsymbol{\lambda}_{p,r}(i|k) \in \mathcal{R}^2$ is the dual variable with respect to (5.6); ρ is the augmented Lagrangian parameter. The coupling collision avoidance constraint (5.5) has been replaced by the non-differential indicator function $I_{\mathcal{C}}(k)$ for $\mathcal{C}_{r_{p,q}}(k)$ as:

$$I_{\mathcal{C}}(k) = \begin{cases} 0, & \text{for } (\Delta\hat{\boldsymbol{r}}_p(i|k), \Delta\hat{\boldsymbol{r}}_q(i|k)) \in \mathcal{C}_{r_{p,q}}(k), \forall e_{p,q}(k) = 1, e_{p,q}(k) \in \mathcal{E}_s(k), \\ \infty, & \text{otherwise.} \end{cases}$$

Given initial values $\boldsymbol{\lambda}^j_{p,r}(i|k)$ and $\Delta\hat{\boldsymbol{r}}^j_p(i|k)$ with $j = 0$, the ADMM decomposition-coordination at each iteration j consists of the following three steps:

Step 1: Each waterborne AGV $p \in \mathcal{V}_s(k)$ solves a local problem with the information $\Delta\hat{\boldsymbol{r}}^j_p(i|k)$ sent from the graph coordinator to update the original perturbation position states $\Delta\boldsymbol{r}_p(i|k)$:

$$\begin{aligned}\left(\Delta\boldsymbol{u}_p^{j+1}(i|k), \Delta\boldsymbol{r}_p^{j+1}(i|k)\right) = \operatorname{argmin} & J_p\left(\boldsymbol{x}_p(k), \Delta\boldsymbol{u}_p(i|k)\right) + \\ & \boldsymbol{\lambda}^j_{p,r}(i|k)^{\mathrm{T}} \left(\Delta\boldsymbol{r}_p(i|k) - \Delta\hat{\boldsymbol{r}}^j_p(i|k)\right) + \rho/2 \left\|\Delta\boldsymbol{r}_p(i|k) - \Delta\hat{\boldsymbol{r}}^j_p(i|k)\right\|_2^2\end{aligned} \tag{5.8}$$

subject to (5.3) – (5.4).

Step 2: The graph coordinator solves the following problem with the information $\Delta \boldsymbol{r}_p^{j+1}(i|k)$ collected from the waterborne AGVs to update the copied perturbation position states $\Delta \hat{\boldsymbol{r}}_p(i|k)$:

$$\Delta \hat{\boldsymbol{r}}_p^{j+1}(i|k) = \operatorname{argmin} I_C(k) + \sum_{p \in \mathcal{V}_s(k)} \Big(\boldsymbol{\lambda}_{p,r}^{j}(i|k)^{\mathrm{T}} \left(\Delta \boldsymbol{r}_p^{j+1}(i|k) - \Delta \hat{\boldsymbol{r}}_p(i|k) \right) + \rho/2 \left\| \Delta \boldsymbol{r}_p^{j+1}(i|k) - \Delta \hat{\boldsymbol{r}}_p(i|k) \right\|_2^2 \Big). \tag{5.9}$$

Step 3: Each waterborne AGV $p \in \mathcal{V}_s(k)$ updates the local dual variables $\boldsymbol{\lambda}_{p,r}(i|k)$ with new information $\Delta \hat{\boldsymbol{r}}_p^{j+1}(i|k)$ from the coordinator as:

$$\boldsymbol{\lambda}_{p,r}^{j+1}(i|k) = \boldsymbol{\lambda}_{p,r}^{j}(i|k) + \rho \left(\Delta \boldsymbol{r}_p^{j+1}(i|k) - \Delta \hat{\boldsymbol{r}}_p^{j+1}(i|k) \right). \tag{5.10}$$

Step 1 and Step 3 are both carried out in parallel on board of each waterborne AGV $p \in \mathcal{V}_s(k)$. The coordinator problem at Step 2 is implemented as Euclidean projections onto $\mathcal{C}_{r_{p,q}}(k)$ implemented as:

$$\Delta \hat{\boldsymbol{r}}_p^{j+1}(k) := \operatorname{argmin} \sum_{p \in \mathcal{V}(k)} \left\| \Delta \hat{\boldsymbol{r}}_p(k) - \left(\Delta \boldsymbol{r}_p^{j+1}(i|k) + \boldsymbol{\lambda}_{p,r}^{j+1}(k)/\rho \right) \right\|_2^2 \tag{5.11}$$

subject to

$$(\Delta \hat{\boldsymbol{r}}_p(i|k), \Delta \hat{\boldsymbol{r}}_q(i|k)) \in \mathcal{C}_{r_{p,q}}(k), \forall e_{p,q}(k) = 1, e_{p,q}(k) \in \mathcal{E}_s(k).$$

Iterations are then alternating between the coordinator and waterborne AGVs until consensus constraints (5.6) are satisfied according to certain criteria, implying that collision avoidance couplings (5.5) are also satisfied on local solutions $\Delta \boldsymbol{r}(i|k)$.

5.3.2 Convergence analysis

To proceed with convergence analysis of the above ADMM iterations, we make the following assumption:

Assumption 5.1 *The centralized subgraph problem* (5.2) – (5.5) *is feasible.*

With Assumption 5.1, the ADMM iterations by Steps 1 – 3 have the following convergence properties:

Proposition 5.1 (Convergence of iterations by Steps 1 – 3) *Under Assumption 5.1, the following convergence is achieved as iteration $j \to \infty$:*

1. *Primal feasibility, i.e., for each waterborne AGV $p \in \mathcal{V}_s(k)$, $\Delta \hat{\boldsymbol{r}}_p^{j}(i|k) \to \Delta \boldsymbol{r}_p^{j}(i|k)$.*
2. *Objective convergence, i.e., primal objective value $J_{\mathcal{G}_s}\left(\boldsymbol{X}_{\mathcal{G}_s}(k), \boldsymbol{U}_{\mathcal{G}_s}(k)\right)$ in* (5.2) *approaches the centralized optimal value $J_{\mathcal{G}_s}^{*}(\boldsymbol{X}_{\mathcal{G}_s}^{(}k), \boldsymbol{U}_{\mathcal{G}_s}(k))$.*

Algorithm 5.1 ADMM based distributed PPF-ATA: processed in parallel by each waterborne AGV $p \in \mathcal{V}_s(k)$

1: Initialize $\boldsymbol{\lambda}_{p,r}^{j}(i|k)$ and $\Delta\hat{\boldsymbol{r}}_{p}^{j}(i|k)$ at $j=0$;
2: **loop**
3: Computes $\Delta\boldsymbol{r}_{p}^{j+1}(i|k)$ solving the problem at Step 1;
4: Sends $\Delta\boldsymbol{r}_{p}^{j+1}(i|k)$ and $\boldsymbol{\lambda}_{p,r}^{j}(i|k)$ to the coordinator;
5: **repeat**
6: Wait;
7: **until** $\Delta\hat{\boldsymbol{r}}_{p}^{j+1}(i|k)$ arrives;
8: Computes $\boldsymbol{\lambda}_{p,r}^{j+1}(i|k)$ as (5.10) at Step 3;
9: $j+1 \rightarrow j$;
10: **end loop**

3. *Dual variable convergence, i.e.,* $\forall p \in \mathcal{V}_s(k)$, $\boldsymbol{\lambda}_{p,r}^{j}(i|k)$ *approaches the optimal dual value* $\boldsymbol{\lambda}_{p,r}^{*}(i|k)$.

Proof: The above proposition follows directly from general ADMM convergence properties in [12] where the proof is established under two mild assumptions: *1*) The (extended-real-valued) separable two functions are closed, proper, and convex. *2*) The unaugmented Lagrangian $\mathcal{L}_0$ has a saddle point. We prove Proposition 5.1 by showing that the two assumptions hold also in our case. Firstly, for each waterborne AGV $p \in \mathcal{V}_s(k)$, define an indicator function $I_{\mathcal{C}_p}(k)$ of the local convex constraint sets $\mathcal{C}_{x_p}(k)$ and $\mathcal{C}_{u_p}(k)$, and $I_{\mathcal{C}_p}(k)=0$ when (5.3) – (5.3) are satisfied and ∞ otherwise. Since local costs J_p are convex, $\mathcal{C}_{x_p}(k)$ and $\mathcal{C}_{u_p}(k)$ are non-empty convex, the problem at Step 1 is solvable. Likewise, indicator function $I_{\mathcal{C}_{p,q}}(k)$ is also closed, proper, and convex, and the problem at Step 2 is also solvable. The first assumption of [12] is satisfied. Secondly, since centralized problem (5.2) – (5.5) is feasible by Assumption 5.1, problem (5.2) – (5.6) is also feasible. Let $\left(\Delta\boldsymbol{r}_{p}^{*}(i|k), \Delta\hat{\boldsymbol{r}}_{p}^{*}(i|k)\right)$ be a feasible solution. As analyzed before, local cost functions J_p are convex and constrained sets $\mathcal{C}_{x_p}(k)$, $\mathcal{C}_{u_p}(k)$, and $\mathcal{C}_{p,q}(k)$ are with non-empty relative interior (Slater's condition holds [11]), there exists $\boldsymbol{\lambda}_{p,r}^{*}(i|k)$ such that $\left(\Delta\boldsymbol{r}_{p}^{*}(i|k), \Delta\hat{\boldsymbol{r}}_{p}^{*}(i|k), \boldsymbol{\lambda}_{p,r}^{*}(i|k)\right)$ is a saddle point of the unaugmented Lagrangian $\mathcal{L}_0(k)$ (set $\rho=0$ for (5.7)), i.e.,

$$\begin{aligned}\mathcal{L}_0\left(\Delta\boldsymbol{r}_{p}^{*}(i|k), \Delta\hat{\boldsymbol{r}}_{p}^{*}(i|k), \boldsymbol{\lambda}_{p,r}(k)\right) &\leqslant \mathcal{L}_0\left(\Delta\boldsymbol{r}_{p}^{*}(i|k), \Delta\hat{\boldsymbol{r}}_{p}^{*}(i|k), \boldsymbol{\lambda}_{p,r}^{*}(k)\right) \\ &\leqslant \mathcal{L}_0\left(\Delta\boldsymbol{r}_{p}(i|k), \Delta\hat{\boldsymbol{r}}_{p}(i|k), \boldsymbol{\lambda}_{p,r}^{*}(k)\right). \end{aligned} \tag{5.12}$$

Therefore, the second assumption of [12] is also satisfied. □

The ADMM based iterations are then implemented in a distributed way alternating between waterborne AGVs $\forall p \in \mathcal{V}_s(k)$ processing in parallel as Algorithm 5.1 and a coordinator waterborne AGV for $\mathcal{G}_s(k)$ processing as Algorithm 5.2. The inputs are initialized at $j=0$. Outputs $\Delta\boldsymbol{u}_{p}^{j}(i|k)$ from Algorithm 5.1 are returned after executing Line 7 in Algorithm 5.2.

Algorithm 5.2 ADMM based distributed PPF-ATA: processed by the coordinator waterborne AGV of $\mathcal{G}_s(k)$

1: **repeat**
2: **repeat**
3: Waits;
4: **until** $\Delta \boldsymbol{r}_p^{j+1}(i|k)$ and $\boldsymbol{\lambda}_{p,r}^{j}(i|k)$ arrive;
5: Computes $\Delta \hat{\boldsymbol{r}}_p^{j+1}(i|k)$ as (5.9) at Step 2;
6: Broadcasts $\Delta \hat{\boldsymbol{r}}_p^{j+1}(i|k)$ to $\forall p \in \mathcal{V}_s(k)$;
7: **until** Stopping criteria are met.

5.3.3 Stopping criteria

Convergence is achieved by primal and dual feasibility which, in practice, are indicated by small primal and dual residuals, respectively, i.e.,

$$r^j(k) = \sum_{p \in \mathcal{V}_s(k)} \left\| \Delta \boldsymbol{r}_p^j(i|k) - \Delta \hat{\boldsymbol{r}}_p^j(i|k) \right\|_2 \leqslant \varepsilon^{\text{pri}}, \tag{5.13}$$

$$s^j(k) = \sum_{p \in \mathcal{V}_s(k)} \left\| \Delta \hat{\boldsymbol{r}}_p^{j+1}(i|k) - \Delta \hat{\boldsymbol{r}}_p^j(i|k) \right\|_2 \leqslant \varepsilon^{\text{dual}}, \tag{5.14}$$

where ε^{pri} and $\varepsilon^{\text{dual}}$ are primal and dual feasibility tolerances specified using an absolute and relative criterion following [12] as:

$$\varepsilon^{\text{pri}} = \sqrt{2n_s N_p}\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \max \left\{ \sum_{p \in \mathcal{V}_s(k)} \left\| \Delta \boldsymbol{r}_p^j(i|k) \right\|_2, \sum_{p \in \mathcal{V}_s(k)} \left\| \Delta \hat{\boldsymbol{r}}_p^j(i|k) \right\|_2 \right\}, \tag{5.15}$$

$$\varepsilon^{\text{dual}} = \sqrt{2n_s N_p}\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \sum_{p \in \mathcal{V}_s(k)} \left\| \boldsymbol{\lambda}_{p,r}^j(k) \right\|_2. \tag{5.16}$$

As suggested in [12], a varying step size ρ^j is implemented as follows to improve practical convergence and make it less dependent on the initial value of ρ:

$$\rho^{j+1} = \begin{cases} 2\rho^j, & \text{for } r^j(k) > 10 s^j(k) \\ \rho^j/2, & \text{for } s^j(k) > 10 r^j(k) \\ \rho^j, & \text{otherwise.} \end{cases} \tag{5.17}$$

Tens of iterations are usually required to achieve convergence to a modest accuracy for conventional ADMM [12]. A variant, fast ADMM, is proposed next to improve the convergence rate in practice.

5.4 Fast ADMM for cooperative distributed waterborne AGVs

Since waterborne AGVs in ADMM are solving local problems selfishly to achieve local control goals and are only coordinated by a penalty term on position variables, the global

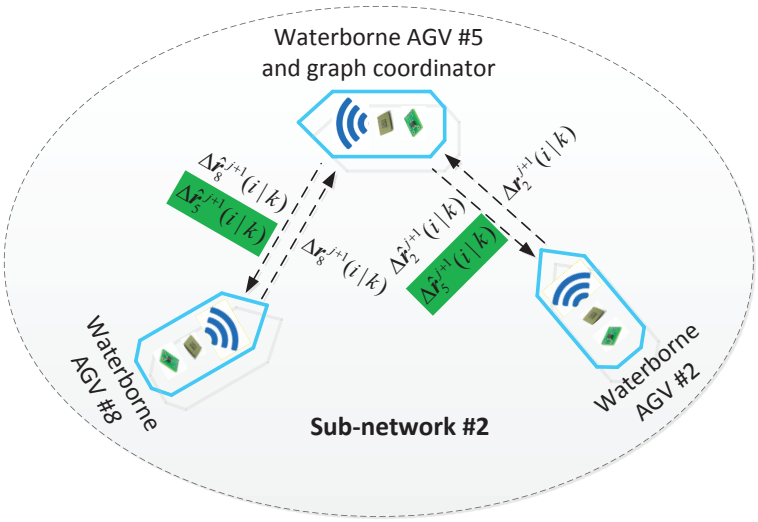


Figure 5.1: Comparison of information flow in fast ADMM.

agreement on the coupling collision avoidance constraints could be slow. The proposed fast ADMM involves adding, iteratively, approximated collision avoidance constraints to local problems based on safe trajectories $\Delta\hat{\boldsymbol{r}}_p^{j+1}(k), p \in \mathcal{V}_s(k)$ from the coordinator as:

$$d_{p,q}^0(i|k) + \boldsymbol{C}(i|k)\Delta r_p(i|k) + \boldsymbol{D}(i|k)\Delta\hat{r}_q^j(i|k) \geqslant D_{\mathrm{s}}, \text{ for } p,q \in \mathcal{V}_s(k), d_{p,q}(k) \leqslant D_{\mathbf{c}}. \quad (5.18)$$

Approximated local collision avoidance constraints (5.18) are different from global collision avoidance constraints (3.19) in two aspects:

1. Waterborne AGV p in (3.19) only cares about the pairwise collision constraints with waterborne AGV q for $p,q \in \mathcal{V}_s(k), e_{p,q} = 1$ with $p < q$, while in (5.18), waterborne AGV p cares about pairwise collision constraints with waterborne AGV q as along as they are in the communication range of each other according to $d_{p,q}(k) \leqslant D_{\mathbf{c}}$; and

2. In (5.18), coupling waterborne AGV q's trajectory is treated as known and fixed while in (3.19), trajectories of both waterborne AGV p and q are variables.

Besides, extra information in fast ADMM needs to be communicated from the coordinator to waterborne AGVs to formulate (5.18). Taking subgraph 2, $\mathcal{G}_2(k) = (\mathcal{V}_2(k), \mathcal{E}_2(k))$ where $\mathcal{V}_2(k) = \{2,5,8\}$ and $e_{2,5} = 1, e_{5,8} = 1$ in Figure 3.2b as an example, the extra communicated information to corresponding waterborne AGVs (waterborne AGV #5 is acting as the coordinator) is highlighted in green in Figure 5.1.

Step 1 in fast ADMM then solves the following problem by each waterborne AGV $p \in \mathcal{V}_s(k)$ parallely updating $\Delta\boldsymbol{r}_p^j(k)$:

$$\begin{aligned}\left(\Delta\boldsymbol{u}_p^{j+1}(i|k), \Delta\boldsymbol{r}_p^{j+1}(i|k)\right) = \operatorname{argmin} & J_p\left(\boldsymbol{x}_p(k), \Delta\boldsymbol{u}_p(i|k)\right) + \\ & \boldsymbol{\lambda}_{p,r}^j(i|k)^{\mathrm{T}}\left(\Delta\boldsymbol{r}_p(i|k) - \Delta\hat{\boldsymbol{r}}_p(i|k)\right) + \rho/2\left\|\Delta\boldsymbol{r}_p(i|k) - \Delta\hat{\boldsymbol{r}}_p^j(i|k)\right\|_2^2\end{aligned} \quad (5.19)$$

subject to (5.3) – (5.4), and (5.18).

Similar with the definitions of $r^j(k)$ and $s^j(k)$ in (5.13) – (5.14), we denote the deviations of $\Delta \boldsymbol{r}_p^{j+1}(i|k)$ from $\Delta \hat{\boldsymbol{r}}_p^j(i|k)$ for all waterborne AGVs in $\mathcal{G}_s$ as $\sigma_s^{j+1}(k)$. Then $\sigma_s^{j+1}(k)$ inherently has two features along with iterations:

1. Deviations $\sigma_s^{j+1}(k)$ cannot be too large because of the augmented penalty term $(\rho/2)$ $\left\|\Delta \boldsymbol{r}_p(i|k) - \Delta \hat{\boldsymbol{r}}_p^j(i|k)\right\|_2^2$ in (5.8) or (5.19), which guarantees local problems at Step 1 of fast ADMM approximate well the original problem at Step 1 before convergence;

2. When both primal and dual residual convergence are achieved satisfying (5.13) – (5.14), i.e., $\Delta \boldsymbol{r}_p^j(i|k) \to \Delta \hat{\boldsymbol{r}}_p^j(i|k)$ and $\Delta \hat{\boldsymbol{r}}_p^{j+1}(i|k) \to \Delta \hat{\boldsymbol{r}}_p^j(i|k)$, deviations $\sigma_s^{j+1}(k) \to 0$ since

$$\begin{aligned} \sigma_s^{j+1}(k) &= \sum_{p \in \mathcal{V}_s(k)} \left(\Delta \boldsymbol{r}_p^{j+1}(i|k) - \Delta \hat{\boldsymbol{r}}_p^j(i|k)\right) \\ &= \sum_{p \in \mathcal{V}_s(k)} \left(\Delta \boldsymbol{r}_p^{j+1}(i|k) - \Delta \hat{\boldsymbol{r}}_p^{j+1}(i|k) + \Delta \hat{\boldsymbol{r}}_p^{j+1}(i|k) - \Delta \hat{\boldsymbol{r}}_p^j(i|k)\right), \end{aligned} \tag{5.20}$$

   which implies that problem at Step 1 of fast ADMM at the terminal iteration finally recovers the original problem at Step 1; feasibility and optimality are satisfied when stopping criteria (5.13) – (5.14) are met.

Remark : Convergence is achieved as Proposition 5.1 for conventional ADMM based distributed PPF-ATA. When ρ is adaptive as (5.17) with iterative information, convergence is difficult to prove. But the varying ρ technique is effective in practice, and convergence can be achieved if ρ becomes fixed after a prior unknown but finite number of iterations [12]. In the case of fast ADMM based distributed PPF-ATA, the approximated local collision avoidance constraints (5.18) are also adaptive with respect to the results from a previous iteration. If the constraint sets formulated by (5.18) become fixed after a finite number of iterations, convergence can also be studied as in [12]. Practical improved convergence rates of fast ADMM are further illustrated and analyzed in Section 5.5.2. □

5.5 Simulation results and discussion

Theoretically, the proposed cooperative distributed approaches apply to any size of waterborne AGV fleets. However, for maritime applications, typical encounters usually involve two or three vehicles. As a representative ITT case study in the port of Rotterdam, we consider a scenario shown as Figure 5.2 with five ITT tasks in Table 5.1 to illustrate the effectiveness of the proposed control approaches. Five waterborne AGVs are denoted as $V_i, i = 1, 2, ..., 5$ each assigned one ITT task[1] and are positioned at their corresponding departure terminals. There are two potential conflicting areas where waterborne AGVs may encounter one another, indicated by the two circles in Figure 5.2. We still use the small scaled marine vehicle model, CSII [121], to represent waterborne AGV dynamics (3.1), (3.5). Simulation results based on CSII are then scaled-up according to Froude scaling law [80], e.g., $1:70$ for length (m) and $1:\sqrt{70}$ for time (s), for the real scale quantities. For simulations in this chapter, sampling time is set as $T_s = 0.5$s and prediction horizon $N_\mathrm{p} = 20$.

[1] The case that more than one ITT tasks are assigned to one waterborne AGV will be considered in Chapter 7.

Table 5.1: ITT tasks for waterborne AGVs.

ITT No.	Routes Berth → Berth	Origin-Destination (m)	Departure-arrival times (s)
$1 \rightarrow V_1$	$1 \rightarrow 2$	$(0, 0) \rightarrow (411, 417)$	$0 \rightarrow 460$
$2 \rightarrow V_2$	$2 \rightarrow 1$	$(411, 417) \rightarrow (0, 0)$	$0 \rightarrow 418$
$3 \rightarrow V_3$	$3 \rightarrow 4$	$(-50, 359) \rightarrow (420, 52)$	$0 \rightarrow 376$
$4 \rightarrow V_4$	$5 \rightarrow 6$	$(1665, 236) \rightarrow (441, 732)$	$0 \rightarrow 920$
$5 \rightarrow V_5$	$7 \rightarrow 8$	$(1121, 1144) \rightarrow (1095, 149)$	$0 \rightarrow 627$

Figure 5.2: ITT scenario for waterborne AGVs.

Parameters for ADMM iterations are: maximum iteration $j_{\max} = 1000$, absolute tolerance $\varepsilon^{\text{abs}} = 10^{-2}$ and relative tolerance $\varepsilon^{\text{rel}} = 5 \times 10^{-4}$. Algorithms are implemented in MATLAB 2014b [75] with optimization problems solved by Cplex [46]. Simulations are run on a platform with Intel(R) Xeon(R) CPU E5-1620 v2 @3.70 GHz.

5.5.1 Safely carrying out ITT tasks

Optimal trajectories from fast ADMM based distributed PPF-ATA control schemes of the five waterborne AGVs are shown in Figure 5.3. All waterborne AGVs are able to track their assigned reference paths well, except for the conflicting areas where some deviations arise to maintain a safety distance away from others. The trade-off between deviations from reference paths and safety has been optimized since all online optimizations are successfully solved. Although trajectories overlay spatially, they do not overlay temporally at the same time, which is demonstrated by positions of waterborne AGVs V_1, V_2, and V_3 at $t = 230$s and waterborne AGVs V_4, V_5 at $t = 418$s plotted in different shapes in Figure 5.3. Collisions are thus successfully avoided. Safety by the fast ADMM based distributed PPF-ATA control approach is further confirmed by Figure 5.4 where Euclidean distances between waterborne AGVs are above the minimum safety distance all the time. Furthermore, waterborne AGVs arrive at their specified destinations punctually at 460s, 423s, 410s, 920s, and 636s, respectively, despite the possibly time consuming behavior for collision avoidance in the conflicting areas.

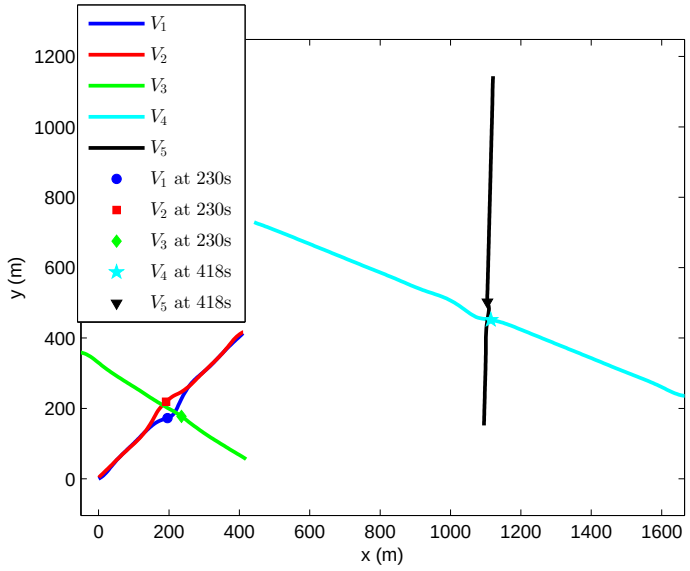


Figure 5.3: Trajectories of five waterborne AGVs.

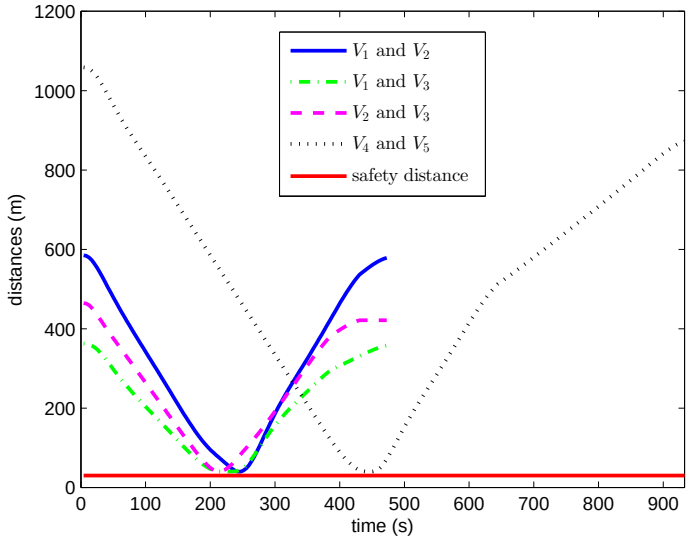


Figure 5.4: Distances between waterborne AGVs.

5.5.2 Convergence behavior in distributed computations

Both the proposed ADMM and fast ADMM based distributed PPF-ATA control approaches are based on the time varying graphs modeled as in Chapter 3. Figure 5.5 shows the graph evolution modeling all working waterborne AGVs controlled by the fast ADMM based distributed PPF-ATA control approach. Circles with numbers represent the numbered waterborne AGVs. Red dashed lines indicate there exists a collision avoidance coupling between the corresponding two waterborne AGVs. Circles with the same color are waterborne AGVs coupled together and thus in one subgraph. Time-varying subgraphs in Figure 5.5 indicate the time-varying couplings among waterborne AGVs.

Based on the above subgraphs, both control approaches can achieve the above overall system performance and safety via iterative decomposition-coordination. However, the fast ADMM based distributed PPF-ATA control scheme has a much faster convergence rate than the ADMM based distributed PPF-ATA controller, as shown in Figure 5.6. The number of iterations with corresponding computation times for ADMM and fast ADMM based controllers are reported in subplot (a) and (b), respectively. Number of iterations are the maximum iterations over all subgraphs. Computation times are the summation of computing times of all iterations of the corresponding subgraph and the computation time per iteration is the summation of the maximum time of all vehicles solving the problem at Step 1, the time solving the problems at Step 2, and Step 3. ADMM involves large numbers of iterations with long computational times in the two conflicting areas. A maximum of over 200 iterations with over 50s is required for ADMM to reach convergence. The computational advantage of fast ADMM over ADMM is obvious. Fast ADMM converges with a maximum of six iterations and 0.27s which is within the system sampling time $T_s = 0.5$s. Also, both subplots show that the number of iterations and the computation times are approximately positively related. Note that in practice, timely feedback within the sampling time is generally critical for real-time control systems. Fast ADMM exploits the algorithmic structure of conventional ADMM computing timely for waterborne AGVs. In more complicated and computational demanding scenarios, besides resorting to tuning controller parameters, improving optimization model structure, and using more powerful computing platforms etc., a reliable decision recovery mechanism is always necessary when real-time decisions cannot be updated timely.

For further comparison, we applied ADMM based distributed PPF-ATA, fast ADMM based distributed PPF-ATA, and centralized controllers to a same subgraph $\mathcal{G}_s(k)$ with $\mathcal{V}_s(k) = \{V_1, V_2, V_3\}$ and $\mathcal{E}_s(k) = \{(V_1, V_2), (V_1, V_3), (V_2, V_3)\}$ at a particular time $t = 142$s. ADMM based distributed PPF-ATA requires 108 iterations with a total solver time of 4.65s before convergence while fast ADMM based distributed PPF-ATA requires only six iterations with 0.27s. Subplot (a), (b), and (c) in Figure 5.7 show the detailed primal residual, dual residual and overall cost convergence behaviors, respectively, of ADMM and fast ADMM based approaches. Primal residuals being small means that trajectories updated in parallel at Step 1 and coordinated trajectories considering collision avoidance at Step 2 are driven close to each other. Dual residuals being small means that trajectories updated by the sub-coordinator do not change much from their previous iteration any more. Global safety and optimality are then achieved when primal and dual residuals satisfy (5.13) – (5.14).

Global optimality is further illustrated in subplot (c) where the centralized cost acts as a baseline and cost differences of ADMM and fast ADMM based controllers iterate approach-

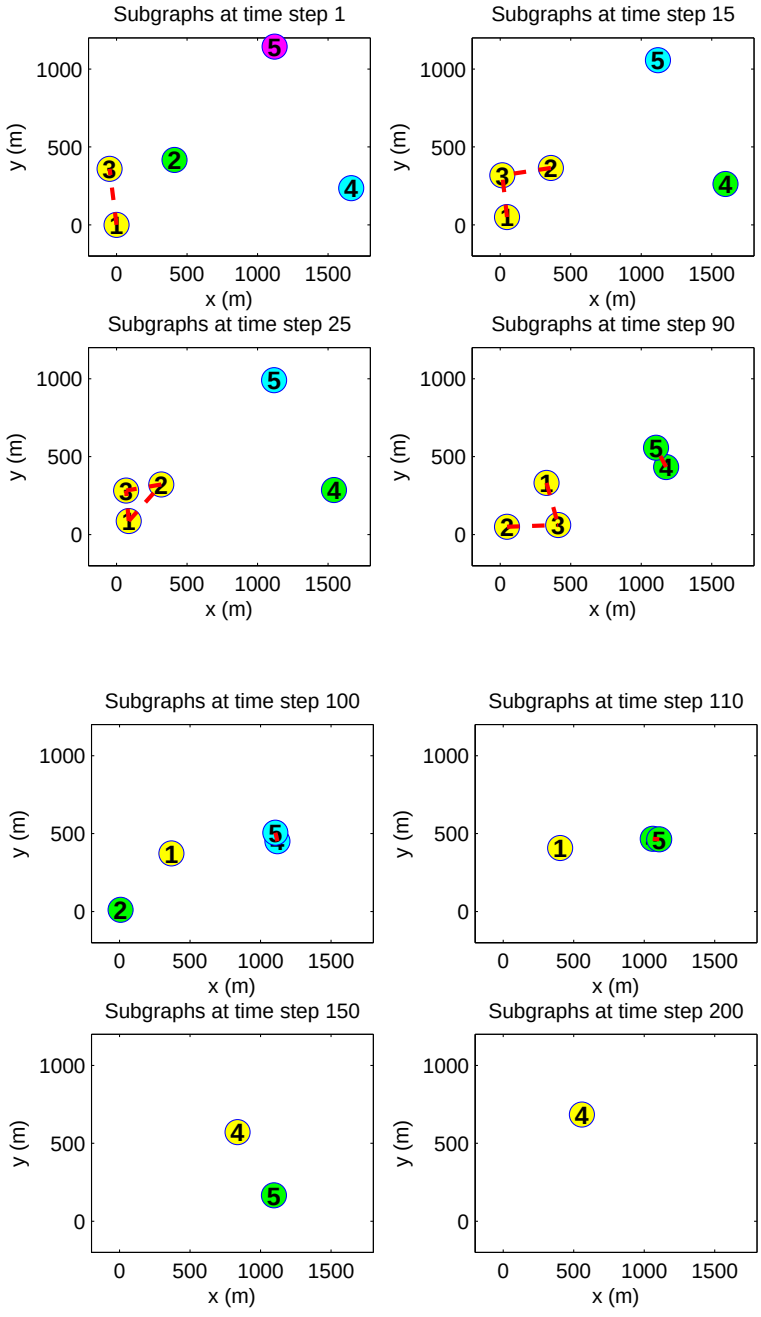


Figure 5.5: Time-varying subgraphs of waterborne AGVs

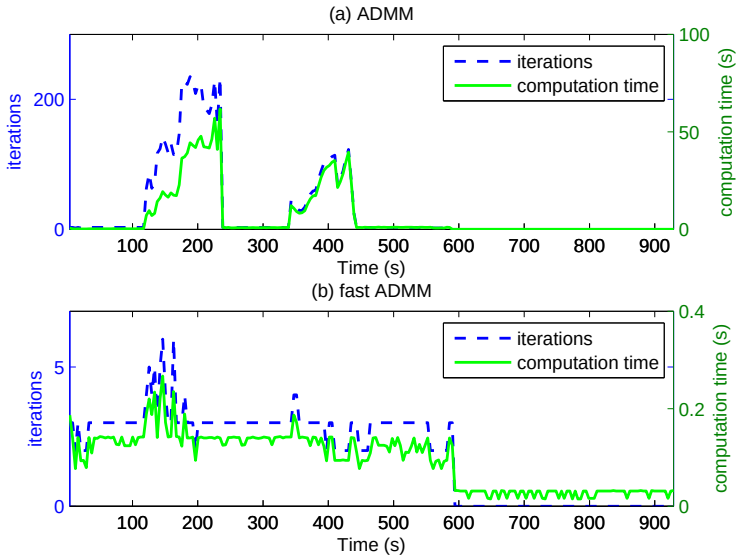


Figure 5.6: Computation time comparison.

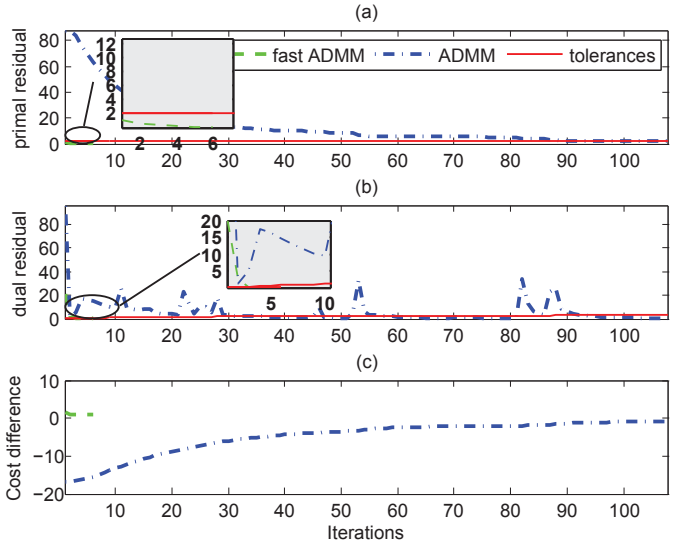


Figure 5.7: Convergence behavior at time $t = 142s$.

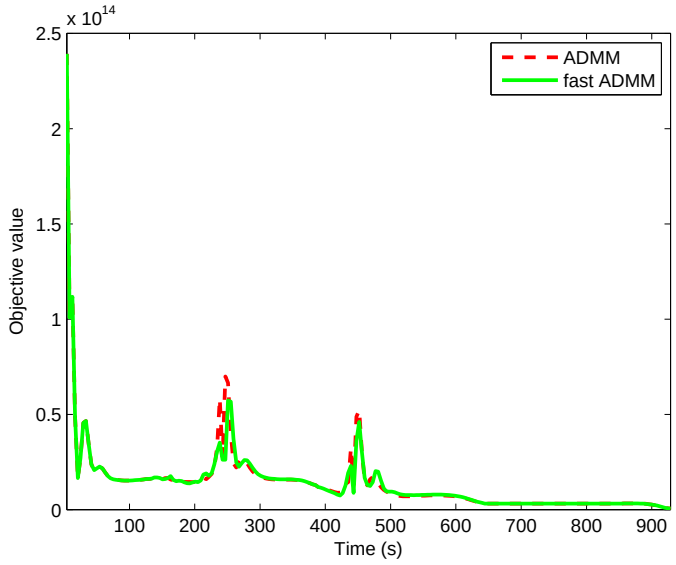


Figure 5.8: Objective comparison.

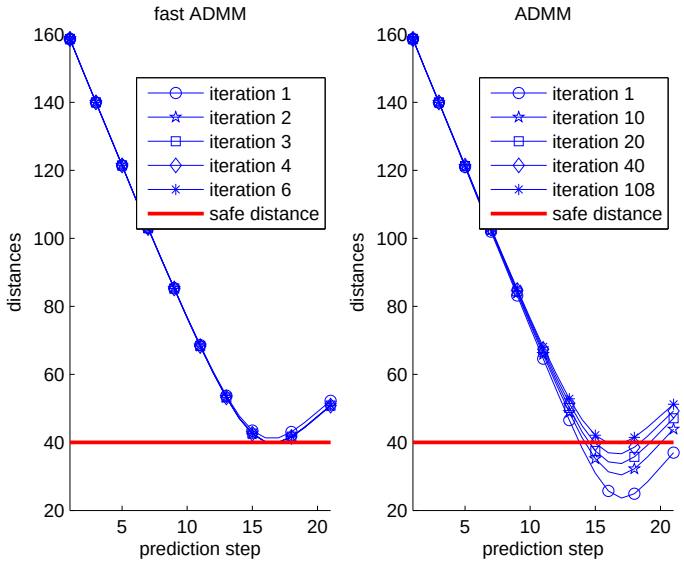


Figure 5.9: Predicted distance evolution at time $t = 142s$.

ing zero. Overall costs of all time steps of the two distributed approaches are further shown as Figure 5.8. Large costs are observed around the initial and two conflicting areas. The two cost trajectories concur although fast ADMM based distributed PPF-ATA approximates the original problem and converges within much fewer iterations. Figure 5.9 further shows how global safety at time $t = 142$s is achieved and compares iterative distances between V_2 and V_3 over the prediction horizon of the two controllers. ADMM based control approach sees distances at several prediction steps below the minimum distance during the first iterations; but by iterative communication and coordination, initial infeasible trajectories are adjusted and driven above the safety line. For fast ADMM based control approach, since waterborne AGVs are not solving local problems selfishly but also considering approximated collision avoidance constraints which could be conservative during first iterations, the initial distances are actually above the safety line. Along with the convergence of primal and dual residuals, however, distances converge to the real collision avoidance constraints yielding global optimality as well.

5.6 Conclusions

In this chapter, we propose a distributed PPF-ATA control approach for multiple waterborne AGVs carrying out ITT tasks. Computing in parallel within time-varying subgraphs as modeled in Chapter 3 is achieved following the ADMM decomposition-coordination procedures. Furthermore, possible poor convergence rates of the conventional ADMM are improved by the proposed fast ADMM based distributed PPF-ATA. In our ITT case study in the port of Rotterdam, comparing ADMM and fast ADMM at the same time step, ADMM requires 108 iterations with a total solver time of 4.65s before convergence while fast ADMM requires only six iterations with 0.27s. Therefore, fast ADMM offers a more practical cooperative distributed approach considering the short sampling time (0.5s in our simulations) of waterborne AGVs. The proposed distributed PPF-ATA algorithm based on fast ADMM achieves cooperative distributed waterborne AGVs and answers the third Key Research Question as listed in Chapter 1.

The distributed control approach in this chapter is applicable to multiple waterborne AGVs when environmental disturbances are perfectly known and the assignment of ITT tasks to waterborne AGVs is given. We will consider cases when environmental disturbances are not perfectly known in Chapter 6 and a closed-loop scheduling and control design in Chapter 7 based on the distributed algorithm proposed in this chapter.

Chapter 6

Cost-effective robust distributed control of waterborne AGVs

In both Chapters 4 and 5, environmental disturbances are assumed to be known perfectly, and waterborne AGV dynamics are modeled as (3.1) and (3.5). In this chapter, we consider multiple waterborne AGVs modeled as (3.1) – (3.2) and (3.6) maneuvering in uncertain environments with not perfectly known disturbances.

The research discussed in this chapter is based on [151, 152]

6.1 Introduction

The problems of a single waterborne AGV carrying out an assigned ITT task and multiple waterborne AGVs with possible collision avoidance couplings are considered in Chapters 4 and 5, respectively. Control goals such as smooth path tracking, timing, energy efficiency, and distributed cooperativeness have been achieved in deterministic scenarios, i.e., perfectly known environmental disturbances. However, in practice, we can only roughly predict the environmental influences by, e.g., weather forecast. For cases in which uncertainties in such predictions exist, maintaining overall safety and energy efficiency in executing ITT tasks with multiple waterborne AGVs remains an issue.

For waterborne AGVs that face uncertainties, besides the control goals in deterministic scenarios, robust satisfaction of system constraints due to physical limitations and collision avoidance is critical. Ideally, it is desirable to have constraint satisfaction for all possible realizations of uncertainties modeled as (3.6) with stochastic characteristics; in practice, however, waterborne AGVs may still fail at those worst cases of the sea which are, though possible, very rare. Moreover, system performance could be degraded dramatically if the system needs to be robust to those cases that rarely happen. Therefore, a practical design should increase safety levels at a cost-effective price.

In this chapter, we propose a cost-effective robust distributed MPC (RDMPC) approach for multiple waterborne AGVs facing uncertain scenarios. The approach is cost-effective in the sense that the overall system robustness level and the associated price of robustness are explicitly optimized considering system and uncertainty characteristics. In particular, we

maximize the robust probability of uncertainties while minimizing the nominal cost with tightened constraints dependent on uncertainty bounds as in tube-based MPC [78]. The problem is still decomposed and coordinated following the Alternating Direction Method of Multipliers (ADMM) achieving cooperative parallel distributed control of coupled waterborne AGVs as in Chapter 5. However, since probabilistic distributions are approximated by introducing binary variables, the convexity assumptions for ADMM convergence do not necessarily hold. Therefore, we propose an efficient integrated branch & bound (B&B) and ADMM algorithm that solves the cost-effective RDMPC problem. The algorithm exploits the special ordered probability sets conducting smart search in B&B and integrates branching criteria with intermediate ADMM results for early termination.

The remainder of this chapter is organized as follows. In Section 6.2, a cost-effective centralized approach is proposed to solve the problem for multiple waterborne AGVs with uncertainties. A linear programming (LP) relaxed RDMPC problem is first formulated and solved in Section 6.3, and the exact RDMPC solutions are proposed in Section 6.4. In Section 6.5, simulation experiments and results are presented and discussed, followed by conclusions of this chapter in Section 6.6.

6.2 Cost-effective robust centralized formulation

In this section, we propose a cost-effective robust centralized approach for coupled waterborne AGVs in one subgraph $\mathcal{G}_s(k) = (\mathcal{V}_s(k), \mathcal{E}_s(k))$ based on tube-based MPC [78]. For notational simplicity, we consider $\mathcal{G}_s(k)$ as the only subgraph in $\mathcal{G}(k)$, and thus use $\mathcal{G}(k) = (\mathcal{V}(k), \mathcal{E}(k))$ for the group of coupled waterborne AGVs considered in this chapter. The centralized control goals are:

- individual ITT task achievement;
- overall safety;
- overall minimal energy consumption; and
- overall cost-effective robustness.

Tube-based MPC [78], as reviewed in Chapter 2, solves closed-loop optimization problems by parameterizing the control policy with an open-loop control sequence and a local feedback. Accordingly, system dynamics with uncertainties are partitioned into a nominal part and an uncertain part. The nominal dynamics generate the tube center and the uncertain dynamics use the local feedback steering uncertain trajectories towards the center formulating tube cross sections. Controlled system trajectories for all possible realizations of uncertainties then evolve around the tube center within the tube. One key advantage of tube MPC is that system goals are achieved by solving a nominal MPC problem with tightened constraints while actual system constraints are still satisfied.

6.2.1 Parameterized uncertainty bounds

One of the assumptions for tube-based MPC is that uncertainties are bounded so that constraints can be properly tightened. However, as modeled by (3.1) – (3.2) and (3.6) in Chap-

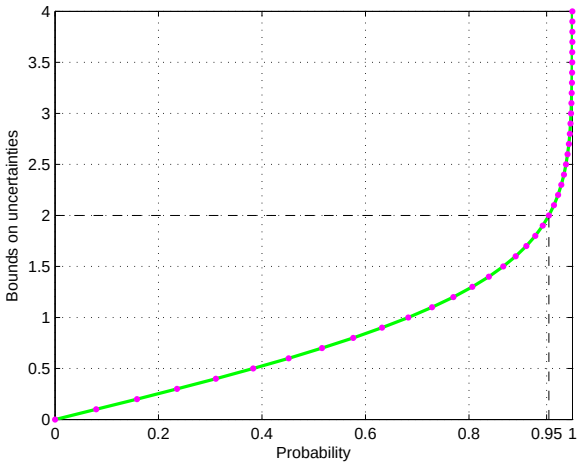


Figure 6.1: Probability-bound of uncertainties following the standard normal distribution.

ter 3, waterborne AGVs moving in open waters experience stochastic environmental uncertainties with infinite support. Obviously, robustness against 100% of such uncertainties is impossible. As shown in Figure 6.1 for a standard normal distribution, the bound on uncertainties increases exponentially as the probability approaches one, and thus could degrade system performance dramatically. A natural way that handles uncertainties with infinite support is to be robust to only a certain probability of uncertainties. For uncertainties $b \sim N(\bar{b}, \Sigma)$ acting upon waterborne AGVs, uncertainty bounds are determined by the inverse Gauss error function erf [1] as

$$z = \bar{b} + \sqrt{2\Sigma}\text{erf}^{-1}(p), \tag{6.1}$$

so that the probability of $b \in [-z, z]$ is p. Figure 6.1 plots (6.1) with $\bar{b} = 0$ and $\Sigma = 1$. System performance and safety depend on the uncertainty bounds z and thus probability p. With large probability p, system performance or even feasibility in finding solutions within physical limits is lost; with small probability p, a certain level of system safety is not guaranteed. We hence define system *robustness level* based on probability p as follows:

Definition 6.1 (Robustness level) *The system with uncertain stochastic uncertainties $b \sim N(\bar{b}, \Sigma)$ is said to have robustness level p if the system is robust to uncertainties in a compact set $[-z, z]$, where z is defined as* (6.1)*.*

A practical design increases the system robustness level at a cost-effective price of being robust. The cost-effective robust approach proposed in [151] uses flexible bounds in tightening constraints and penalizes deviations of the corresponding probability from a desired robustness level. The idea is to integrate and make explicit use of the known probabilistic distributions relating probability and uncertainty bounds in online optimizations. Online optimizations can become intractable using complex distribution functions as Gauss error

functions. We, therefore, approximate the distribution functions by:

$$p = \boldsymbol{a}\boldsymbol{P}, \tag{6.2}$$

$$z = \bar{b} + \sqrt{\Sigma}\,(\boldsymbol{a}\boldsymbol{Z}), \tag{6.3}$$

$$\sum \boldsymbol{a} = 1, \boldsymbol{a} \in \{0,1\}^{n_b} \tag{6.4}$$

where $\boldsymbol{Z} \in \mathcal{R}^{n_b}$ and $\boldsymbol{P} \in \mathcal{R}^{n_b}$ are bound and probability vectors, respectively, and satisfy $\boldsymbol{P} = \text{erf}(\boldsymbol{Z}/\sqrt{2})$; $\boldsymbol{a}$ is a binary vector that parametrizes the distribution function by relating the probability and bound via (6.2) and (6.3), and guaranteeing that exactly one probability or bound is selected via (6.4). The set of red dots in Figure 6.1 is an example of $\boldsymbol{P}$ and corresponding $\boldsymbol{Z}$ with $n_b = 40$; the selected (p,z) pair with $p = 0.95$ and $z = 2$ is an example of the 21st element of $\boldsymbol{a}$ being one and all others being zero. For parametrized uncertainties (6.2) – (6.4) with unknown bounds, we are now ready to design a tube-based robust MPC control strategy for multiple waterborne AGVs.

6.2.2 Tube-based MPC for successively linearized models with paramenterized bounded uncertainties

We consider the successively linearized uncertain waterborne AGV models (3.13) – (3.15), (3.18). The uncertain system states of each waterborne AGV $p \in \mathcal{V}(k)$ are partitioned as

$$\boldsymbol{x}_p(i|k) = \boldsymbol{x}_p^0(i|k) + \Delta\bar{\boldsymbol{x}}_p(i|k) + \Delta\tilde{\boldsymbol{x}}_p(i|k), \tag{6.5}$$

where $\Delta\bar{\boldsymbol{x}}_p$ is the nominal perturbation state and $\Delta\tilde{\boldsymbol{x}}_p$ the deviation of the actual perturbation state $\Delta\boldsymbol{x}_p$ (defined as (3.13)) from $\Delta\bar{\boldsymbol{x}}_p$. Likewise, the control input is partitioned as:

$$\boldsymbol{u}_p(i|k) = \boldsymbol{u}_p^0(i|k) + \Delta\bar{\boldsymbol{u}}_p(i|k) + \Delta\tilde{\boldsymbol{u}}_p(i|k). \tag{6.6}$$

The nominal system dynamics then evolve as:

$$\bar{\boldsymbol{x}}_p(i+1|k) = \boldsymbol{x}_p^0(i|k) + \boldsymbol{A}_p(i|k)\Delta\bar{\boldsymbol{x}}_p(i|k) + \boldsymbol{B}_p(i|k)\Delta\bar{\boldsymbol{u}}_p(i|k)$$

and the uncertain dynamics evolve as:

$$\Delta\tilde{\boldsymbol{x}}_p(i+1|k) = \boldsymbol{A}_p(i|k)\Delta\tilde{\boldsymbol{x}}_p(i|k) + \boldsymbol{B}_p(i|k)\Delta\tilde{\boldsymbol{u}}_p(i|k) + \boldsymbol{E}_p(i|k)\Delta b(i|k), \tag{6.7}$$

with $\Delta\tilde{\boldsymbol{x}}_p(0|k) = \boldsymbol{0}$ since $\bar{\boldsymbol{x}}_p(0|k) = \boldsymbol{x}_p^0(0|k) = \boldsymbol{x}_p(k)$. The certain part of b, i.e., $\bar{b}$ has been incorporated in $\boldsymbol{x}_p^0(i|k)$ in calculating $\boldsymbol{x}_p^0$ and thus in the nominal dynamics. By (6.3), the uncertain part is then bounded as $\Delta b(i|k) \in \mathcal{W}(i|k)$ where $\mathcal{W}(i|k)$ is a time-varying compact set with origin in its interior explicitly dependent on $\boldsymbol{a}(i|k)$ as:

$$\mathcal{W}(i|k) =: \left[-\sqrt{\Sigma}\,(\boldsymbol{a}(i|k)\boldsymbol{Z}), \sqrt{\Sigma}\,(\boldsymbol{a}(i|k)\boldsymbol{Z})\right]. \tag{6.8}$$

Incorporating feedback in predictions and optimizing over control policies rather than control sequences is necessary when uncertainties exist to reduce conservativeness. However, optimizing over arbitrary feedback policies is practically intractable. Similarly as tube-based MPC [78], we employ an affine feedback control policy as:

$$\Delta\tilde{\boldsymbol{u}}_p(i|k) = \boldsymbol{K}_p(i|k)\Delta\tilde{\boldsymbol{x}}_p(i|k), \tag{6.9}$$

where $\boldsymbol{K}_p(i|k)$ is a time-varying feedback gain that needs to be calculated online. Particularly, for time-varying uncertain dynamics (6.7), we solve $\boldsymbol{K}_p(i|k)$ as a finite horizon unconstrained time-varying LQR controller [111] as:

- set $\boldsymbol{P}_p(N_\mathrm{p}|k) := \boldsymbol{Q}_f$;
- for $i = N_\mathrm{p}, N_\mathrm{p}-1..., 1$

$$\begin{aligned}\boldsymbol{P}_p(i-1|k) =& \boldsymbol{Q} + \boldsymbol{A}_p^{\mathrm{T}}(i|k)\boldsymbol{P}_p(i|k)\boldsymbol{B}_p(i|k)\left(\boldsymbol{R}+\boldsymbol{B}_p^{\mathrm{T}}(i|k)\right.\\ &\left.\boldsymbol{P}_p(i|k)\boldsymbol{B}_p(i|k)\right)^{-1}\boldsymbol{B}_p^{\mathrm{T}}(i|k)\boldsymbol{P}_p(i|k)\boldsymbol{A}_p(i|k);\end{aligned} \tag{6.10}$$

- for $i = 0, 1, ..., N_\mathrm{p}-1$

$$\boldsymbol{K}_p(i|k) = -\left(\boldsymbol{R}+\boldsymbol{B}_p(i|k)^{\mathrm{T}}\boldsymbol{P}_p(i|k)\boldsymbol{B}_p(i|k)\right)^{-1}\boldsymbol{B}_p(i|k)^{\mathrm{T}}\boldsymbol{P}_p(i+1|k)\boldsymbol{A}_p(i|k), \tag{6.11}$$

where $\boldsymbol{Q}$, $\boldsymbol{Q}_f$ and $\boldsymbol{R}$ are state cost, terminal state cost, and input cost matrices, respectively, of the time-varying LQR controller, being the same for all waterborne AGVs. Then (6.7) in closed-loop is:

$$\Delta\tilde{\boldsymbol{x}}_p(i+1|k) = \boldsymbol{A}_{\boldsymbol{K}_p}(i|k)\Delta\tilde{\boldsymbol{x}}_p(i|k) + \boldsymbol{E}_p(i|k)\Delta b(i|k) \tag{6.12}$$

with $\boldsymbol{A}_{\boldsymbol{K}_p}(j|k) = \boldsymbol{A}_p(i|k) + \boldsymbol{B}_p(i|k)\boldsymbol{K}_p(i|k)$. Denote the set for uncertain perturbation states as $\tilde{\mathcal{X}}_p$, i.e., $\Delta\tilde{\boldsymbol{x}}_p(i|k) \in \tilde{\mathcal{X}}_p(i|k)$, we further have:

$$\tilde{\mathcal{X}}_p(i+1|k) := \boldsymbol{A}_{\boldsymbol{K}_p}(i|k)\tilde{\mathcal{X}}_p(i|k) \oplus \boldsymbol{E}_p(i|k)\,\mathcal{W}(a(i|k)) \tag{6.13}$$

with $\tilde{\mathcal{X}}_p(0|k) = \{\boldsymbol{0}\}$. The operator $\oplus$ defines the Minkowski set sum: $\mathcal{A}\oplus\mathcal{B} := \{a+b|a \in \mathcal{A}, b \in \mathcal{B}\}$. The sizes of sets $\{\tilde{\mathcal{X}}_p(i|k)\}$ are expected to be smaller than those calculated from unstable pairs of $(\boldsymbol{A}(i|k), \boldsymbol{B}(i|k))$ since $\boldsymbol{A}_{\boldsymbol{K}}(i|k)$ is now stable by design. Moreover, $\{\tilde{\mathcal{X}}_p(i|k)\}$ explicitly depends on the bounds of uncertainty sets $\mathcal{W}(i|k)$, and thus is also parametrized by $a(i|k)$. On the one hand, system performance desires small uncertainty sets; on the other hand, it is necessary the system robustness level approaches one.

Following (6.5) and (6.13), the state tube with $\{\boldsymbol{x}_p^0(i|k) + \Delta\tilde{\boldsymbol{x}}_p(i|k)\}$ as centre and $\{\tilde{\mathcal{X}}_p(i|k)\}$ as cross sections is defined as:

$$\mathcal{X}_p(i|k) := \left(\boldsymbol{x}_p^0(i|k) + \Delta\tilde{\boldsymbol{x}}_p(i|k)\right) \oplus \tilde{\mathcal{X}}_p(i|k), \tag{6.14}$$

and likewise control input tube is defined as:

$$\mathcal{U}_p(i|k) := \left(\boldsymbol{u}_p^0(i|k) + \Delta\bar{\boldsymbol{u}}_p(i|k)\right) \oplus \boldsymbol{K}_p(i|k)\tilde{\mathcal{X}}_p(i|k). \tag{6.15}$$

System constraints imposed by (3.3) are local convex constraints on states and control inputs, indicated by $\mathcal{C}_{x_p}$ and $\mathcal{C}_{u_p}$, respectively, for waterborne AGV p. Coupling collision avoidance constraints (3.8) between waterborne AGV p and q are non-convex and are later convexified as (3.19) indicated by $\mathcal{C}_{x_{p,q}}$. Then, the centralized problem, termed *Problem 1*, for multiple waterborne AGVs is readily formulated to achieve the goals listed at the beginning of this section as:

$$\min_{\Delta\bar{\boldsymbol{u}},\boldsymbol{a}} \sum_{p\in\mathcal{V}(k)} J_p\left(\boldsymbol{x}_p(k), \Delta\bar{\boldsymbol{u}}_p(i|k)\right) + Q_a\left\|\boldsymbol{1} - p(i|k)\right\|_1 \tag{6.16}$$

subject to

$$\mathcal{X}_p(i|k) \subseteq \mathcal{C}_{x_p} \qquad \forall p \in \mathcal{V}(k), \tag{6.17}$$
$$\mathcal{U}_p(i|k) \subseteq \mathcal{C}_{u_p} \qquad \forall p \in \mathcal{V}(k), \tag{6.18}$$
$$(\mathcal{X}_p(i|k), \mathcal{X}_q(i|k)) \subseteq \mathcal{C}_{x_{p,q}} \qquad \forall e_{p,q} = 1, \tag{6.19}$$
$$p(i|k) = \boldsymbol{a}(i|k)\boldsymbol{P}, \tag{6.20}$$
$$\sum \boldsymbol{a}(i|k) = 1, \boldsymbol{a}(i|k) \in \{0,1\}^{n_b} \tag{6.21}$$

where $J_p\left(\boldsymbol{x}_p(k), \Delta\bar{\boldsymbol{u}}_p(i|k)\right) = \left\|\boldsymbol{x}_p^e(k)\right\|_{\boldsymbol{w_1}}^2 + \|\Delta\boldsymbol{u}_p(k)\|_{\boldsymbol{w_2}}^2 + \left\|\boldsymbol{x}_p^{\boldsymbol{v}}(k)\right\|_{w_3}^2$ is the nominal local convex cost function. The three terms in J_p are formulated so that path following errors, control input changes, and kinetic energy consumption are minimized as in Chapter 4 to achieve smooth path tracking, energy efficiency, and arrival time awareness for ITT. Cost function (6.16) also penalizes the deviation of the overall robustness level from one in the last term where $\boldsymbol{Q}_a$ is a weight parameter.

6.2.3 Implementations

Solving *Problem 1* requires set computations $\oplus$ which are time consuming, and due to the time-varying nature of the system (3.13) – (3.15), (3.18) and constraint (3.19), set computations are necessarily conducted online. Therefore, implemented tube-based MPC treats each constraint separately. Suppose that each waterborne AGV $p \in \mathcal{V}(k)$ has I_x state constraints and I_u control input constraints. For $j = 1, ..., I_x$,

$$\boldsymbol{C}_p^j(i|k)\boldsymbol{x}_p(i|k) = \boldsymbol{C}_p^j(i|k)\left(\boldsymbol{x}_p^0(i|k) + \Delta\bar{\boldsymbol{x}}_p(i|k) + \Delta\tilde{\boldsymbol{x}}_p(i|k)\right) \leqslant d_p^j(i|k).$$

It is sufficient to tighten the nominal term via an offset defined as the out-bounding of the uncertainty term:

$$\begin{aligned} B_{x_p}^j(i|k) &= \max_{\boldsymbol{a}(i|k)} \left\{ \boldsymbol{C}_p^j(i|k)\Delta\tilde{\boldsymbol{x}}_p(i|k) \middle| \Delta\tilde{\boldsymbol{x}}_p(i|k) \in \tilde{\mathcal{X}}_p(i|k) \right\} \\ &= \max_{\boldsymbol{a}(i|k)} \left\{ \boldsymbol{C}_p^j(i|k)\boldsymbol{\Theta}_p(i|k)\boldsymbol{\Phi}_p(i|k) | \Delta b(i|k) \in \mathcal{W}(i|k) \right\} \end{aligned} \tag{6.22}$$

with

$$\boldsymbol{\Theta}_p(i|k) = \left[\prod_{n=1}^{i-1} \boldsymbol{A}_{K_p}(n|k)\boldsymbol{E}_p(0|k) \;\; \cdots \;\; \prod_{n=i-1}^{i-1} \boldsymbol{A}_{K_p}(n|k)\boldsymbol{E}_p(i-2|k) \;\; \boldsymbol{E}_p(i-1|k) \right]$$

and $\boldsymbol{\Phi}(i|k) = \left[\Delta b^{\mathrm{T}}(0|k) \;\; \Delta b^{\mathrm{T}}(1|k) \;\; \cdots \;\; \Delta b^{\mathrm{T}}(i-1|k)\right]^{\mathrm{T}}$. For general compact sets defined as (6.8) with bounds $\sqrt{\Sigma}(\boldsymbol{a}(i|k)\boldsymbol{Z})$, a solution to (6.22) is guaranteed to exist [67]. Moreover, for the structured norm bounded uncertainties in our case, explicit maximization based on the duality norm [67] is applicable as:

$$\begin{aligned} B_{x_p}^j(i|k) &= \left\| \boldsymbol{C}_p^j(i|k)\boldsymbol{\Theta}_p(i|k)\boldsymbol{T}(i) \right\|_1 \\ &= \max_{\Delta\boldsymbol{b}_t(i|k)} \left\{ \boldsymbol{C}_p^j(i|k)\boldsymbol{\Theta}_p(i|k)\boldsymbol{T}(i)\Delta\boldsymbol{b}_t(i|k) | \|\Delta\boldsymbol{b}_t(i|k)\|_\infty \leqslant 1 \right\}, \end{aligned} \tag{6.23}$$

where $\Delta \boldsymbol{b}_t(i|k) = \boldsymbol{T}^{-1}(i)\Delta \boldsymbol{b}(i|k)$ and $\boldsymbol{T}(i)$ is a diagonal translation matrix

$$\boldsymbol{T}(i+1) = \begin{bmatrix} \boldsymbol{T}(i) & \boldsymbol{0} \\ \boldsymbol{0} & \sqrt{\Sigma}(\boldsymbol{a}(i|k)\boldsymbol{Z}) \end{bmatrix} \tag{6.24}$$

The possibility of explicit maximization avoids solving the programming problem (6.22) for each constraint online which is then tightened as:

$$\boldsymbol{C}_p^j(i|k)\Delta\bar{\boldsymbol{x}}_p(i|k) \leqslant d_p^j(i|k) - \boldsymbol{C}_p^j(i|k)\boldsymbol{x}_p^0(i|k) - B_{x_p}^j(i|k). \tag{6.25}$$

Control input constraints (6.18) are dealt with in a similar way. For $j = 1, ..., I_u$:

$$\boldsymbol{H}_p^j(i|k)\boldsymbol{u}_p(i|k) = \boldsymbol{H}_p^j(i|k)\left(\boldsymbol{u}_p^0(i|k) + \Delta\bar{\boldsymbol{u}}_p(i|k) + \Delta\tilde{\boldsymbol{u}}_p(i|k)\right) \leqslant h_p^j$$

Tightened constraints on nominal control inputs can be obtained as:

$$\boldsymbol{H}_p^j(i|k)\Delta\bar{\boldsymbol{u}}_p(i|k) \leqslant h_p^j(i|k) - \boldsymbol{H}_p^j(i|k)\boldsymbol{u}_p^0(i|k) - B_{u_p}^j(i|k). \tag{6.26}$$

with tightening offsets

$$B_{u_p}^j(i|k) = \left\|\boldsymbol{H}_p^j(i|k)\boldsymbol{K}_p(i|k)\boldsymbol{\Theta}_p(i|k)\boldsymbol{T}(i)\right\|_1. \tag{6.27}$$

Suppose there are I_c coupling constraints in (6.19), and for $j = 1, ..., I_c$,

$$\begin{bmatrix} \boldsymbol{E}_p^j(i|k) & \boldsymbol{E}_q^j(i|k) \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_p(i|k) \\ \boldsymbol{r}_q(i|k) \end{bmatrix} \leqslant f_{p,q}(i|k), \tag{6.28}$$

where $p \in \mathcal{V}(k), q \in \mathcal{V}(k)$ and $e_{p,q} = 1$. Then (6.28) can also be tightened as in (6.25) but involving tube information (nominal trajectory and bounds) from both coupled waterborne AGVs p and q. Tightened constraints are on nominal perturbation position states $\Delta\bar{\boldsymbol{r}}_p(i|k)$ and $\Delta\bar{\boldsymbol{r}}_q(i|k)$ as:

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{E}_p^j(i|k) & \boldsymbol{E}_q^j(i|k) \end{bmatrix} \begin{bmatrix} \Delta\bar{\boldsymbol{r}}_p(i|k) \\ \Delta\bar{\boldsymbol{r}}_q(i|k) \end{bmatrix} \leqslant f_{p,q}(i|k) - \\ &\begin{bmatrix} \boldsymbol{E}_p^j(i|k) & \boldsymbol{E}_q^j(i|k) \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_p^0(i|k) \\ \boldsymbol{r}_q^0(i|k) \end{bmatrix} - B_p^j(i|k) - B_q^j(i|k), \end{aligned} \tag{6.29}$$

with tightening offsets

$$B_p^j(i|k) = \left\|\boldsymbol{E}_p^j(i|k)\boldsymbol{\Theta}_p(i|k)\boldsymbol{T}(i)\right\|_1 \tag{6.30}$$

and

$$B_q^j(i|k) = \left\|\boldsymbol{E}_q^j(i|k)\boldsymbol{\Theta}_q(i|k)\boldsymbol{T}(i)\right\|_1. \tag{6.31}$$

Define the tightened state, control input, and collision avoidance constraint sets as $\mathcal{C}_{x_p}(\boldsymbol{a}(k))$, $\mathcal{C}_{u_p}(\boldsymbol{a}(k))$, and $\mathcal{C}_{r_{p,q}}(\boldsymbol{a}(k))$, respectively, then constraints (6.17), (6.18), and (6.19) are implemented as

$$\Delta\bar{\boldsymbol{x}}_p(i|k) \in \mathcal{C}_{x_p}(\boldsymbol{a}(k)) \qquad \forall p \in \mathcal{V}(k), \tag{6.32}$$

$$\Delta\bar{\boldsymbol{u}}_p(i|k) \in \mathcal{C}_{u_p}(\boldsymbol{a}(k)) \qquad \forall p \in \mathcal{V}(k), \tag{6.33}$$

$$(\Delta\bar{\boldsymbol{r}}_p(i|k), \Delta\bar{\boldsymbol{r}}_q(i|k)) \in \mathcal{C}_{r_{p,q}}(\boldsymbol{a}(k)) \qquad \forall e_{p,q} = 1, \tag{6.34}$$

Note that compared with (5.3) – (5.5) for the deterministic case in Chapter 5, (6.32) – (6.34) depend explicitly on $\boldsymbol{a}(k)$. *Problem 1* is a mixed integer programming (MIP) problem that involves $N_{\mathrm{p}} n_b$ binary variables. A sensible simplification is to fix the probability over N_{p} with n_b binary variables and to replace (6.21) by

$$\sum \boldsymbol{a}(k) = 1, \boldsymbol{a}(k) \in \{0,1\}^{n_b}. \tag{6.35}$$

The implemented centralized problem with objective (6.16) and constraints (6.32), (6.33), (6.34), and (6.35) is termed *Problem 2*. *Problem 2* is coupled due to

- pairwise collision avoidance constraints (6.34);
- coupling cost for overall robustness level.

Chapter 5 proposes a distributed approach based on ADMM for multiple waterborne AGVs in deterministic cases. However, due to the introduction of binary variables $\boldsymbol{a}(k)$, the convexity assumptions for ADMM convergence do not necessarily hold. A relaxed cost-effective RDMPC algorithm is proposed first in the next section for the centralized *Problem 2*; exact solutions are proposed in Section 6.4.

Remark : The cost-effective robustness in *Problem 2* could be interpreted as a type of "relax-penalize" soft-constrained approach with constraint violations being parametrized explicitly by uncertainty distributions. This is important since constraint violations are penalized considering the intrinsic interactions between system and uncertainty properties instead of being penalized equally. For stochastically distributed uncertainties, penalizing already very small deviations of probability to the desirable one could dramatically degrade system performance and even cause controller failures. In other words, it is not cost-effective to sacrifice unnecessarily large system performance to increase little robustness level. □

6.3 Relaxed cost-effective RDMPC with ADMM

The main advantage of ADMM is that parallel distributed decision making still achieves overall feasibility and optimality. However, the cost-effective robust centralized *Problem 2* involves binary variables in both the cost function and constraints, which does not necessarily guarantee convergence of ADMM iterations. In this section, we propose a relaxed cost-effective RDMPC problem using ADMM. This problem will act as the node problem in an efficient B&B search procedure to be proposed in Section 6.4.

6.3.1 Derivation of ADMM for the relaxed RDMPC problem

The binary variable constraints (6.35) of the original centralized *Problem 2* are first relaxed as:

$$\sum \boldsymbol{a}(k) = 1, 0 \leqslant \boldsymbol{a}(k) \leqslant 1. \tag{6.36}$$

The relaxed cost-effective problem with (6.36) is termed *Problem 3*. Besides the collision avoidance constraints (6.34) as in the deterministic case, waterborne AGVs with uncertainties need to reach consensus also on an overall robustness level depending on $\boldsymbol{a}(k)$.

The same as that in Chapter 5, one of the waterborne AGVs in graph $\mathcal{G}(k)$ is assigned as the graph coordinator. For each waterborne AGV $p \in \mathcal{V}(k)$, we introduce a copy of the bound variables as $\hat{\boldsymbol{a}}_p(k)$ and a copy of the perturbation nominal position variables as $\Delta\hat{\boldsymbol{r}}_p(i|k)$. Then *Problem 3* can be rewritten as:

$$\min_{\Delta\bar{\boldsymbol{u}}(i|k),\boldsymbol{a}'_p(i|k)} \sum_{p\in\mathcal{V}(k)} \left(J_p\left(\boldsymbol{x}_p(k),\Delta\bar{\boldsymbol{u}}_p(i|k)\right)+Q_a/n_v\left\|1-\hat{\boldsymbol{a}}_p(k)\boldsymbol{P}\right\|_1\right) \tag{6.37}$$

subject to $\forall p \in \mathcal{V}(k)$

$$\Delta\bar{\boldsymbol{x}}_p(i|k) \in \mathcal{C}_{x_p}(\hat{\boldsymbol{a}}_p(k)), \tag{6.38}$$

$$\Delta\bar{\boldsymbol{u}}_p(i|k) \in \mathcal{C}_{u_p}(\hat{\boldsymbol{a}}_p(k)), \tag{6.39}$$

$$\sum\hat{\boldsymbol{a}}_p(k) = 1, 0 \leqslant \hat{\boldsymbol{a}}_p(k) \leqslant 1, \tag{6.40}$$

$$(\Delta\hat{\boldsymbol{r}}_p(i|k),\Delta\hat{\boldsymbol{r}}_q(i|k)) \in \mathcal{C}_{r_{p,q}}(\hat{\boldsymbol{a}}_p(k),\hat{\boldsymbol{a}}_q(k)), \tag{6.41}$$

$$\hat{\boldsymbol{a}}_p(k) = \boldsymbol{a}(k), \tag{6.42}$$

$$\Delta\hat{\boldsymbol{r}}_p(i|k) = \Delta\bar{\boldsymbol{r}}_p(i|k), \tag{6.43}$$

where convex constraint sets $\mathcal{C}_{x_p}(\hat{\boldsymbol{a}}_p(k))$, $\mathcal{C}_{u_p}(\hat{\boldsymbol{a}}_p(k))$, and $\mathcal{C}_{r_{p,q}}(\hat{\boldsymbol{a}}_p(k),\hat{\boldsymbol{a}}_q(k))$ now explicitly depend on local copies $\hat{\boldsymbol{a}}_p(k)$ of $\boldsymbol{a}(k)$ due to local translation matrices instead of (6.24):

$$\boldsymbol{T}_p(i+1) = \begin{bmatrix} \boldsymbol{T}_p(i) & \boldsymbol{0} \\ \boldsymbol{0} & \sqrt{\Sigma}\left(\hat{\boldsymbol{a}}_p(i|k)\boldsymbol{Z}\right) \end{bmatrix}.$$

The augmented Lagrangian [12] that relaxes (6.42) and (6.43) is:

$$\mathcal{L}_\rho(k) = \sum_{p\in\mathcal{V}(k)} \left(J_p\left(\boldsymbol{x}_p(k),\Delta\bar{\boldsymbol{u}}_p(i|k)\right)+Q_a/n_v\left\|1-\hat{\boldsymbol{a}}_p(k)\boldsymbol{P}\right\|_1 + \begin{bmatrix} \boldsymbol{\lambda}_{p,a}(k)^{\mathrm{T}} & \boldsymbol{\lambda}_{p,r}(i|k)^{\mathrm{T}} \end{bmatrix} \right. \tag{6.44}$$

$$\left. \begin{bmatrix} \boldsymbol{a}(k)-\hat{\boldsymbol{a}}_p(k) \\ \Delta\bar{\boldsymbol{r}}_p(i|k)-\Delta\hat{\boldsymbol{r}}_p(i|k) \end{bmatrix} + \rho/2\left\| \begin{bmatrix} \boldsymbol{a}(k)-\hat{\boldsymbol{a}}_p(k) \\ \Delta\bar{\boldsymbol{r}}_p(i|k)-\Delta\hat{\boldsymbol{r}}_p(i|k) \end{bmatrix} \right\|_2^2 \right) + I_{\mathcal{C}}(k),$$

where $\boldsymbol{\lambda}_{p,a}(k) \in \mathcal{R}^{n_b}$ and $\boldsymbol{\lambda}_{p,r}(i|k) \in \mathcal{R}^2$ are dual variables with respect to (6.42) and (6.43), respectively; ρ is the augmented Lagrangian parameter. The coupling collision avoidance constraint (6.41) has been replaced by the non-differential indicator function $I_{\mathcal{C}}$ for $\mathcal{C}_{r_{p,q}}(\hat{\boldsymbol{a}}_p(k),\hat{\boldsymbol{a}}_q(k))$ as:

$$I_{\mathcal{C}}(k) = \begin{cases} 0, & \text{for (6.41)} \\ \infty, & \text{otherwise.} \end{cases}$$

Given initial values $\boldsymbol{\lambda}^j_{p,a}(k)$, $\boldsymbol{\lambda}^j_{p,r}(i|k)$, $\boldsymbol{a}^j(k)$, and $\Delta\hat{\boldsymbol{r}}^j_p(i|k)$ at $j=1$, the ADMM decomposition-coordination at each iteration j consists of the following three steps:

Step 1: each waterborne AGV $p \in \mathcal{V}(k)$ solves the following local problem updating copied uncertainty bounds $\hat{\boldsymbol{a}}_p(k)$ and original perturbation position states $\Delta\bar{\boldsymbol{r}}_p(k)$:

$$\left(\Delta\bar{\boldsymbol{u}}^{j+1}_p(i|k),\hat{\boldsymbol{a}}^{j+1}_p(k),\Delta\boldsymbol{r}^{j+1}_p(i|k)\right) = \operatorname{argmin} J_p + Q_a/n_v\left\|1-\hat{\boldsymbol{a}}_p(k)\boldsymbol{P}\right\|_1 +$$

$$\begin{bmatrix} \boldsymbol{\lambda}^j_{p,a}(k)^{\mathrm{T}} & \boldsymbol{\lambda}^j_{p,r}(i|k)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} -\hat{\boldsymbol{a}}_p(k) \\ \Delta\bar{\boldsymbol{r}}_p(i|k) \end{bmatrix} + \rho/2\left\| \begin{bmatrix} \boldsymbol{a}^j(k)-\hat{\boldsymbol{a}}_p(k) \\ \Delta\bar{\boldsymbol{r}}_p(i|k)-\Delta\hat{\boldsymbol{r}}^j_p(i|k) \end{bmatrix} \right\|_2^2 \tag{6.45}$$

subject to (6.38) – (6.40).

Step 2: the graph coordinator solves the following problem updating original uncertainty bound variables $\boldsymbol{a}(k)$ and copied perturbation position states $\Delta\hat{\boldsymbol{r}}_p(i|k)$:

$$\left(\boldsymbol{a}^{j+1}(k),\Delta\hat{\boldsymbol{r}}_p^{j+1}(i|k)\right) = \operatorname{argmin} I_C(k) + \sum_{p\in\mathcal{V}(k)} \left(\begin{bmatrix} \boldsymbol{\lambda}_{p,a}^{j}(k)^{\mathrm{T}} & \boldsymbol{\lambda}_{p,r}^{j}(i|k)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}(k) \\ -\Delta\hat{\boldsymbol{r}}_p(i|k) \end{bmatrix} \right. \tag{6.46}$$

$$\left. +\rho/2 \left\| \begin{bmatrix} \boldsymbol{a}(k) - \hat{\boldsymbol{a}}_p^{j+1}(k) \\ \Delta\bar{\boldsymbol{r}}_p^{j+1}(i|k) - \Delta\hat{\boldsymbol{r}}_p(i|k) \end{bmatrix} \right\|_2^2 \right).$$

Step 3: each waterborne AGV $p \in \mathcal{V}(k)$ updates local dual variables $\boldsymbol{\lambda}_{p,a}(k)$ and $\boldsymbol{\lambda}_{p,r}(i|k)$ as:

$$\begin{bmatrix} \boldsymbol{\lambda}_{p,a}^{j+1}(i|k) \\ \boldsymbol{\lambda}_{p,r}^{j+1}(i|k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda}_{p,a}^{j}(k) \\ \boldsymbol{\lambda}_{p,r}^{j}(k) \end{bmatrix} + \rho \begin{bmatrix} \boldsymbol{a}^{j+1}(k) - \hat{\boldsymbol{a}}_p^{j+1}(k) \\ \Delta\bar{\boldsymbol{r}}_p^{j+1}(i|k) - \Delta\hat{\boldsymbol{r}}_p^{j+1}(i|k) \end{bmatrix}. \tag{6.47}$$

Step 1 and Step 3 are both carried out in parallel on board of each waterborne AGV $p \in \mathcal{V}(k)$. The coordinator problem at Step 2 can be split further into two problems:

1. A global variable consensus problem [12] implemented as:

$$\boldsymbol{a}^{j+1}(k) := 1/n_v \sum_{p\in\mathcal{V}(k)} \left(\hat{\boldsymbol{a}}_p^{j+1}(k) + \boldsymbol{\lambda}_{p,a}^{j}(k)/\rho\right), \tag{6.48}$$

2. A problem of determining the Euclidean projections onto $\mathcal{C}_{r_{p,q}}(\hat{\boldsymbol{a}}_p(k),\hat{\boldsymbol{a}}_q(k))$ implemented as:

$$\Delta\hat{\boldsymbol{r}}_p^{j+1}(i|k) := \operatorname{argmin} \sum_{p\in\mathcal{V}(k)} \left\| \Delta\hat{\boldsymbol{r}}_p(i|k) - \left(\Delta\bar{\boldsymbol{r}}_p^{j+1}(i|k) + \boldsymbol{\lambda}_{p,r}^{j+1}(i|k)/\rho\right) \right\|_2^2 \tag{6.49}$$

subject to (6.41).

Note that in addition to the updated information on bound uncertainties and position states, local tightening offsets (6.30) and (6.31) are necessary for the coordinator to formulate $\mathcal{C}_{r_{p,q}}(\hat{\boldsymbol{a}}_p(k),\hat{\boldsymbol{a}}_q(k))$ at Step 2. Iterations are then alternating between the coordinator and waterborne AGVs until consensus constraints (6.42) and (6.43) are satisfied, illustrated as Figure 6.2, implying that

- all waterborne AGVs have agreed on the uncertainty bound parametrized by $\boldsymbol{a}(k)$; and
- local solutions are also satisfying collision avoidance coupling (6.34).

6.3.2 Convergence analysis and stopping criteria

Similar to the convergence analysis in Chapter 5 for deterministic scenarios, convergence analysis of the ADMM-based distributed algorithm for the relaxed cost-effective RDMPC problem is based on an assumption for the relaxed centralized problem:

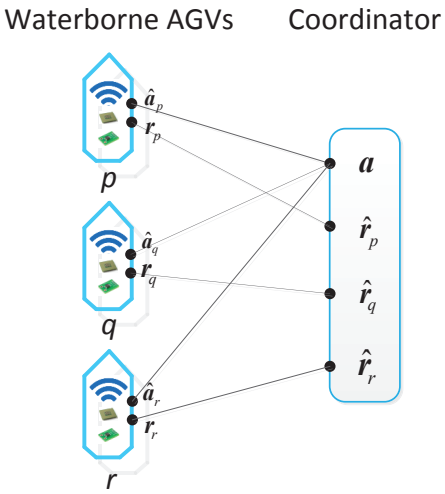


Figure 6.2: Agreement between waterborne AGVs and the coordinator.

Assumption 6.1 *The relaxed cost-effective robust centralized problem Problem 3 is feasible.*

Then the relaxed cost-effective RDMPC problem has the following convergence properties:

Proposition 6.1 (Convergence of the relaxed cost-effective RDMPC) *Under Assumption 6.1, the following convergence is achieved as iteration $j \to \infty$:*

1. *Primal feasibility, i.e., for each waterborne AGV $p \in \mathcal{V}(k)$, $\hat{\boldsymbol{a}}_p^j(k) \to \boldsymbol{a}^j(k)$ and $\Delta\hat{\boldsymbol{r}}_p^j(i|k) \to \Delta\bar{\boldsymbol{r}}_p^j(i|k)$.*

2. *Objective convergence, i.e., primal objective value of* (6.37) *approaches optimality.*

3. *Dual variable convergence, i.e., $\forall p \in \mathcal{V}(k)$, $\boldsymbol{\lambda}_{p,a}^j(k)$ and $\boldsymbol{\lambda}_{p,r}^j(k)$ approach dual optimal points $\boldsymbol{\lambda}_{p,a}^*(k)$ and $\boldsymbol{\lambda}_{p,r}^*(k)$, respectively.*

The proof of Proposition 6.1 follows the proof for Proposition 5.1, and is skipped in this chapter.

Stopping criteria can also be extended from those in Chapter 5 as:

$$r^j(k) = \sum_{p\in\mathcal{V}(k)} \left\| \begin{bmatrix} \boldsymbol{a}^j(k) - \hat{\boldsymbol{a}}_p^j(k) \\ \Delta\bar{\boldsymbol{r}}_p^j(i|k) - \Delta\hat{\boldsymbol{r}}_p^j(i|k) \end{bmatrix} \right\|_2 \leqslant \varepsilon^{\text{pri}}, \tag{6.50}$$

$$s^j(k) = \sum_{p\in\mathcal{V}(k)} \left\| \rho \begin{bmatrix} \boldsymbol{a}^{j+1}(k) - \boldsymbol{a}^j(k) \\ \Delta\hat{\boldsymbol{r}}_p^{j+1}(i|k) - \Delta\hat{\boldsymbol{r}}_p^j(i|k) \end{bmatrix} \right\|_2 \leqslant \varepsilon^{\text{dual}}, \tag{6.51}$$

Algorithm 6.1 Relaxed cost-effective RDMPC: processed in parallel by each waterborne AGV $p \in \mathcal{G}(k)$

initializes $\boldsymbol{\lambda}_{p,a}^{j}(k)$, $\boldsymbol{\lambda}_{p,r}^{j}(i|k)$, $\boldsymbol{a}^{j}(k)$, and $\Delta\hat{\boldsymbol{r}}_{p}^{j}(i|k)$ at $j=1$;
loop
  computes $\hat{\boldsymbol{a}}_{p}^{j+1}(k), \Delta\boldsymbol{r}_{p}^{j+1}(i|k)$ as (6.45);
  sends $\hat{\boldsymbol{a}}_{p}^{j+1}(k), \Delta\boldsymbol{r}_{p}^{j+1}(i|k)$, $\boldsymbol{\lambda}_{p,a}^{j}(k)$ and $\boldsymbol{\lambda}_{p,r}^{j}(i|k)$ to the coordinator;
  repeat
    wait;
  until $\boldsymbol{a}^{j+1}(k), \Delta\hat{\boldsymbol{r}}_{p}^{j+1}(k)$ arrive;
  computes $\boldsymbol{\lambda}_{p,a}^{j+1}(k)$ and $\boldsymbol{\lambda}_{p,r}^{j+1}(i|k)$ as (6.47);
  $j+1 \rightarrow j$;
end loop

Algorithm 6.2 Relaxed cost-effective RDMPC: processed by the coordinator waterborne AGV of $\mathcal{G}(k)$

repeat
  repeat
    wait;
  until $\hat{\boldsymbol{a}}_{p}^{j+1}(k), \Delta\boldsymbol{r}_{p}^{j+1}(i|k)$, $\boldsymbol{\lambda}_{p,a}^{j}(k)$ and $\boldsymbol{\lambda}_{p,r}^{j}(i|k)$ arrive;
  computes $\boldsymbol{a}^{j+1}(k), \Delta\hat{\boldsymbol{r}}_{p}^{j+1}(i|k)$ as (6.46);
  broadcasts $\boldsymbol{a}^{j+1}(k), \Delta\hat{\boldsymbol{r}}_{p}^{j+1}(i|k)$ to $\forall p \in \mathcal{V}(k)$;
until stopping criteria (6.50) – (6.51) are met.

with ε^{pri} and $\varepsilon^{\text{dual}}$ defined as:

$$\varepsilon^{\text{pri}} = \sqrt{2n_v N_p + n_b}\,\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \max \left\{ \sum_{p\in\mathcal{V}(k)} \left\| \begin{bmatrix} \hat{\boldsymbol{a}}_{p}^{j}(k) \\ \Delta\bar{\boldsymbol{r}}_{p}^{j}(i|k) \end{bmatrix} \right\|_2, \sum_{p\in\mathcal{V}(k)} \left\| \begin{bmatrix} \boldsymbol{a}^{j}(k) \\ \Delta\hat{\boldsymbol{r}}_{p}^{j}(i|k) \end{bmatrix} \right\|_2 \right\},$$

$$\varepsilon^{\text{dual}} = \sqrt{2n_v N_p + n_b}\,\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \sum_{p\in\mathcal{V}(k)} \left\| \begin{bmatrix} \boldsymbol{\lambda}_{p,a}^{j}(k) \\ \boldsymbol{\lambda}_{p,r}^{j}(i|k) \end{bmatrix} \right\|_2.$$

Algorithms 5.1 and 5.2 are extended as Algorithms 6.1 and 6.2 processed by AGVs and the graph coordinator, respectively. However, output $\boldsymbol{a}^{j}(k)$ from the relaxed problem generally contains fractional elements, which is infeasible for the original *Problem 2*. We next exploit the problem structure and propose an efficient integrated B&B and ADMM based cost-effective RDMPC method to retrieve exact solutions.

6.4 Efficient exact cost-effective RDMPC

In this section, we derive an efficient integrated B&B and ADMM algorithm that retrieves exact solutions from the Relaxed problem as presented in the previous section. Cooperative

distributed computations as well as overall feasibility, optimality and cost-effective robustness are still achieved. The approach exploits special ordered sets (SOS) [6] conducting smart search in B&B, and integrating branching criteria with intermediate ADMM results for early termination of iterations.

6.4.1 B&B with SOS1

Generic B&B is an exact solution paradigm that is the core for virtually any modern software solving MIP problems [141]. B&B uses a search tree to implicitly enumerate possible solutions by recursively partitioning the solution space into subspaces and pruning solution subspaces that preclude the optimal solution. Active tree nodes are subspace problems that are relaxed and solved as Algorithm 6.1 and 6.2 to provide lower bounds (for minimization problems) that are then compared with the current upper bound, i.e., the incumbent objective. Denote the incumbent objective as J^*, the current tree node objective J^*_{LP} with an optimal solution $\left(\Delta\bar{\boldsymbol{u}}^*_p(i|k), \boldsymbol{a}^*(k)\right)$, then three cases arise after solving the relaxed cost-effective RDMPC problem:

- Case 1: $J^*_{\mathrm{LP}} \geqslant J^*$, then prune the node;
- Case 2: $J^*_{\mathrm{LP}} \leqslant J^*$ and $\boldsymbol{a}^*(k)$ is integer, then set $J^* = J^*_{\mathbf{LP}}$, $\left(\Delta\bar{\boldsymbol{u}}^*_p(i|k), \boldsymbol{a}^*(k)\right)$ as the incumbent solution, and prune the node;
- Case 3: $J^*_{\mathrm{LP}} \leqslant J^*$ and $\boldsymbol{a}^*(k)$ is fractional, then generate descendants of the current node.

Branches in first two cases are both pruned either by bound or by optimality. The third case requires partitioning the current solution space or subspace further into subspaces generating descendant nodes. Generic B&B branches on fractional variables $a_i, i \in I$ with index set $I = \{1, 2, ..., n_b\}$ by adding branching constraints as $a_i = 0$ and $a_i = 1$ to the two descendant nodes, respectively. Search strategies such as depth first, breadth first or best first etc. [141] then decides the order in which the active nodes are to be processed by Algorithm 6.1 and 6.2 and matching the three cases to either obtain a better solution or verify the optimality of the current feasible solution. In such procedures, each variable may be branched on multiple times during the search. Moreover, since the constraints on binary variables are in the form of (6.35), the branch with $a_i = 0$ makes little progress in improving the bounds while the branch with $a_i = 1$ immediately returns an integer solution; the search tree is unbalanced. We next exploit the structure of the model and branch on set of variables to have balanced search trees.

Constraints as (6.35) are called generalized upper bound (GUB) constraints [141] where exactly one variable takes value 1 and all others are 0. Instead of branching on variables, branching on the GUB constraints with $\sum_{i \in I'} a_i = 1$ for one subspace and $\sum_{i \in I'} a_i = 0$ the other results in a more balanced search tree; $I' \subseteq I$ is a subset of binary variable indices. In *Problem 2*, constraints (6.35) enforce that exactly one bound or one probability is selected for the uncertainties. Furthermore, since a_i with increasing index i corresponds to a larger probability as (6.2) or larger bound as (6.3), there is a logical ordering of variables $a_i, i \in I$ which are called special ordered sets of type 1 (SOS1) first proposed in [6]. The vectors deciding on the logical ordering, i.e., $\boldsymbol{P}$ or $\boldsymbol{Z}$, are termed the *reference row*. The corresponding

branching method is then called SOS1 branching. Since we put heavy penalty on deviations of the robustness level p to one, the probability vector $\boldsymbol{P}$ is set as the reference row in our case. The idea is to take advantage of the conventionally overlooked fractional solutions in the third case of B&B and combine them with the reference row to branch in a smarter way.

Suppose the current solution $\boldsymbol{a}^*(k)$ to the relaxed cost-effective RDMPC problem is fractional and satisfies the SOS1 constraint (6.35):

$$a_1^* + a_2^* + ... + a_{n_b}^* = 1 \tag{6.52}$$

Since the reference row $\boldsymbol{P}$ is ordered as $p_1 \leqslant p_2 \leqslant ... \leqslant p_{n_b}$,the SOS1 branching suggests the branch point as:

$$i^* := \underset{i \in I \ominus I'}{\operatorname{argmin}} \{p_i\}, \tag{6.53}$$

where the index set I' is computed as:

$$I' := \{i | p_i \leqslant \sum_{i \in I} a_i^* p_i\}. \tag{6.54}$$

The solution space can now be partitioned into two subspaces by adding $\sum_{i \in I'} a_i = 1$ for one subspace and $\sum_{i \in I'} a_i = 0$ or equivalently, $\sum_{i \in I \ominus I'} a_i = 1$ for the other subspace. Generally, when the relaxed problems at the two subspaces are solved, integer feasible solutions are obtained either for the branch with $\sum_{i \in I'} a_i = 1$ as $a_{\max_i\{I'\}} = 1$ or for the branch with $\sum_{i \in I \ominus I'} a_i = 0$ as $a_{\min_i\{I \ominus I'\}} = 1$. If not, the subspace with fractional solutions can apply the above SOS1 branching procedure recursively.

6.4.2 Integrated B&B with SOS1 and ADMM

Besides the special structure SOS1 constraints that facilitate efficient branching, two observations are noteworthy for further improving the efficiency of the B&B search procedure with intermediate results of ADMM iterations.

First, Algorithms 6.1 and 6.2 need not necessarily run to convergence for the lower bound J_{LP}^*. In B&B, J_{LP}^* is compared with the incumbent objective J^* in three branching cases. However, we also save primal and dual objectives before convergence that can be used for the comparisons to achieve early termination of ADMM as well as early pruning of a B&B node. Define J_{P}^j as the primal overall objective of (6.37) and J_{D}^j the dual overall Lagrangian objective $\mathcal{L}_0^j(k)$. Then, we track an intermediate objective value J_{LP}^j at iteration j as:

- $J_{\mathrm{LP}}^j = J_{\mathrm{P}}^j$ if primal feasibility (6.50) and dual feasibility (6.51) are not satisfied;
- $J_{\mathrm{LP}}^j = \max\{J_{\mathrm{P}}^j, J_{\mathrm{D}}^j\}$ if dual feasibility (6.51) is satisfied but primal feasibility (6.50) is not satisfied;
- $J_{\mathrm{LP}}^j = J_{\mathrm{LP}}^*$ if both primal (6.50) and dual feasibility (6.51) are satisfied.

The following proposition is then established for safe early termination of ADMM and pruning of the node.

Proposition 6.2 (Early termination and pruning) *At iteration j of a node problem, i.e., Problem 3 solved by Algorithm 6.1 and 6.2, if $J_{LP}^{j} \geqslant J^{*}$, then we can safely terminate ADMM iterations and prune the node.*

Proof : Proposition 6.2 follows from the fact that $J_{\mathrm{LP}}^{j} \leqslant J_{\mathrm{LP}}^{*}$ where J_{LP}^{*} is a lower bound of *Problem 2*. Whenever $J_{\mathrm{LP}}^{j} \geqslant J^{*}$, we also have $J_{\mathrm{LP}}^{*} \geqslant J^{*}$, therefore, iterations of ADMM can be safely stopped and the current node can be pruned as Case 1 of B&B. If primal feasibility is not achieved, the primal objective J_{P}^{j} of a minimization problem with feasible solutions by solving partially constrained problems at Step 1 is smaller than the optimal objective, i.e., J_{LP}^{*}. If dual feasibility is achieved, since the dual problem is a maximization problem of which the optimal dual value $J_{\mathrm{D}}^{*} \geqslant J_{\mathrm{D}}^{j}$ provides an lower bound for J_{LP}^{*}, i.e., $J_{\mathrm{D}}^{*} \leqslant J_{\mathrm{LP}}^{*}$, we have $J_{\mathrm{D}}^{j} \leqslant J_{\mathrm{LP}}^{*}$. Proposition 6.2 is established. □

Secondly, we observe that if the solution for binary variables are integer, the consensus constraints (6.42) are satisfied within a small number of iterations; otherwise if fractional, they stay fractional for possible large number of iterations before convergence. This motivates a special treatment of the stopping conditions in ADMM. Specifically, for the second case of B&B, if $\hat{\boldsymbol{a}}_{\boldsymbol{p}}^{\boldsymbol{j}}$ are integer and the stopping criteria for consensus constraint (6.42) are satisfied, the local variables $\hat{\boldsymbol{a}}_{\boldsymbol{p}}^{\boldsymbol{j}}$ are set as $\tilde{\boldsymbol{a}}^{\boldsymbol{j}}$ during all the following iterations until the stopping criteria for consensus constraint (6.42) are also satisfied. If, however, $\boldsymbol{a}_{\boldsymbol{p}}^{\boldsymbol{j}}$ are fractional, since the solution space of this node is to be further partitioned, an optimal objective value J_{LP}^{*} and an optimal solution $\left(\Delta\bar{\boldsymbol{u}}_{p}^{*}(i|k), \boldsymbol{a}^{*}(k)\right)$ are not candidates for the ultimate exact optimal solution anyway; instead, we are only interested in fractional solutions that could provide collective information together with the reference row $\boldsymbol{P}$ to suggest a branching point as (6.53) and (6.54). Therefore, we can safely terminate the ADMM iterations with a small $j'_{\max}$. The third case of B&B then follows the SOS1 branching procedure using fractional solutions $\tilde{\boldsymbol{a}}^{\boldsymbol{j}}$ which are not necessarily optimal.

For waterborne AGVs coupled as $\mathcal{G}(k) = (\mathcal{V}(k), \mathcal{E}(k))$, we then provide an efficient exact cooperative distributed solution for the centralized cost-effective robust *Problem 2*. Waterborne AGVs $\forall p \in \mathcal{V}(k)$ still process in parallel as Algorithm 6.1. The waterborne AGV coordinator of $\mathcal{G}(k)$ processes the computations of J_{LP}^{j} and three branching rules in B&B using intermediate ADMM results. Algorithm 6.2 in this way is extended into Algorithm 6.3.

Algorithm 6.3 Exact cost-effective RDMPC by the coordinator waterborne AGV of $\mathcal{G}(k)$

1: Initialize $J^{*} = \infty$; mark the root node as active;
2: **repeat**
3: **repeat**
4: **repeat**
5: Wait;
6: **until** $\hat{\boldsymbol{a}}_{p}^{j+1}(k), \Delta\boldsymbol{r}_{p}^{j+1}(i|k)$, $\boldsymbol{\lambda}_{p,a}^{j}(k)$ and $\boldsymbol{\lambda}_{p,r}^{j}(i|k)$ arrive;
7: Computes $\boldsymbol{a}^{j+1}(k), \Delta\hat{\boldsymbol{r}}_{p}^{j+1}(i|k)$ as (6.46);
8: Broadcasts $\boldsymbol{a}^{j+1}(k), \Delta\hat{\boldsymbol{r}}_{p}^{j+1}(i|k)$ to $\forall p \in \mathcal{V}(k)$;
9: **if** J_{LP}^{j+1} is computed according to (6.50) – (6.51) **then**

if Case 1: $J_{\mathrm{LP}}^{j+1} \geqslant J^*$ **then**w
  prune the node and go to Line 1 of Algorithm 6.1;
end if
if Case 2: $J_{\mathrm{LP}}^{j+1} \leqslant J^*$ and $\boldsymbol{a}_p^j$ for $p \in \mathcal{V}(k)$ are integer and equal **then**
  set $\boldsymbol{a}_p^{j+1} = \hat{\boldsymbol{a}}^{j+1}$
end if
if Case 3: $J_{\mathrm{LP}}^{j+1} \leqslant J^*$ and $\hat{\boldsymbol{a}}^{j+1}$ is fractional and $j = j'_{\max}$ **then**
  Mark the two descendants as active and go to Line 1 of Algorithm 6.1;
end if
end if
until (6.50) – (6.51) are met.
Set $J^* = J_{\mathrm{LP}}^j$ and $\left(\Delta\bar{\boldsymbol{u}}_p^j(i|k), \tilde{\boldsymbol{a}}^j(k)\right)$ as the incumbent solution;
until no active nodes

Table 6.1: ITT tasks for waterborne AGVs

ITT tasks	Routes Berth → Berth	Origin-Destination (m)	Departure-arrival times (s)
$1 \rightarrow V_1$	$1 \rightarrow 2$	(0, 0) → (411, 417)	$0 \rightarrow 401$
$2 \rightarrow V_2$	$2 \rightarrow 1$	(411, 417) → (0, 0)	$0 \rightarrow 401$
$3 \rightarrow V_3$	$3 \rightarrow 4$	(-50, 359) → (420, 52)	$0 \rightarrow 401$

6.5 Simulation results and discussion

In this section, simulations are run to demonstrate the effectiveness of the proposed cost-effective RDMPC strategy for waterborne AGVs that carry out ITT tasks while facing with uncertainties. The proposed algorithm is applicable to any size of waterborne AGV fleets. For simplicity, we consider scenarios that could illustrate how the algorithm works. Figure 6.3 shows the ITT scenario in the port of Rotterdam with three ITT tasks in Table 6.1 assigned to three waterborne AGVs which are denoted as V_1, V_2, and V_3. The proposed cost-effective RDMPC algorithm is also applicable to scenarios with more waterborne AGVs handling more ITT tasks. In Figure 6.3, there is one potential conflicting area where waterborne AGVs may encounter one another, as the yellow circle in Figure 6.3 shows. The port authority has predicted environmental disturbances but stochastic uncertainties exist in the predictions. Waterborne AGVs are required to track the shortest reference path between the origin and destination smoothly and arrive at the destination as punctual as possible. Moreover, each waterborne AGV computes locally while achieves overall energy efficiency and avoids collisions with others despite of environmental disturbances due to wind, waves, and currents.

We assume homogeneous dynamics (3.1)–(3.2), (3.6) for three waterborne AGVs with identical values for hydrodynamic parameters based on a small-scaled vessel model, CSII [121]. Our simulation uses a sampling time of $T_s = 0.6$ s with a prediction horizon $N_{\mathrm{p}} = 7$. Other controller parameters are set as follows: weight parameters $\boldsymbol{w}_1 = \mathrm{diag}([100, 50, 10])$, $\boldsymbol{w}_2 = \boldsymbol{I}_{3\times 3}$, $\boldsymbol{w}_3 = 1$, and $\boldsymbol{Q}_p = 10^7$. Time-varying LQR parameters in (6.10) – (6.11) are set

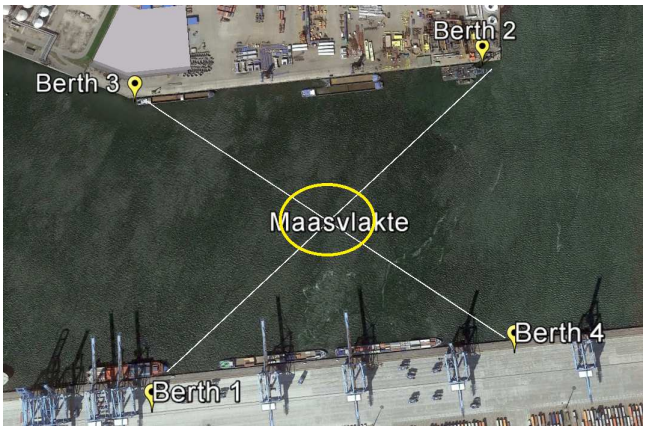


Figure 6.3: ITT scenario for waterborne AGVs.

as: $\boldsymbol{Q} = \text{diag}(\begin{bmatrix} 100 & 20 & 20 & 0 & 0 & 0 \end{bmatrix})$, $\boldsymbol{Q}_f = \text{diag}(\begin{bmatrix} 10^4 & 10^4 & 10^4 & 0 & 0 & 0 \end{bmatrix})$, $\boldsymbol{R} = \boldsymbol{I}_{3\times 3}$. Stopping criteria for ADMM convergence are set with absolute tolerance $\varepsilon^{\text{abs}} = 2\times 10^{-3}$ and relative tolerance $\varepsilon^{\text{rel}} = 10^{-4}$. Stochastic uncertain predictions of environmental disturbances from the port authority are $b \sim N(\bar{b}, \Sigma)$ with mean value $\bar{\boldsymbol{b}} = 343000\text{N}$ and variances $\boldsymbol{\Sigma} = 219520\text{N}^2$. Physical system constraints are: -0.5 m/s $\leqslant u \leqslant 1.8$ m/s and

$$\boldsymbol{\tau}_{\max} = -\boldsymbol{\tau}_{\min} = \begin{bmatrix} 1372000\text{N} & 1372000\text{N} & 72030000\text{Nm} \end{bmatrix}^{\text{T}}.$$

The bound vector $\boldsymbol{Z} \in \mathcal{R}^n$ with $n_b = 21$ is sampled with interval 0.2 from $[0, 3.9]$ since for standard normal distributions, erf($\frac{3.9}{\sqrt{2}}$)=0.9999. Then the probability vector $\boldsymbol{P} = \text{erf}(\frac{3.9}{\sqrt{2}})$. We further assign a big value $M = 10^4$ and a probability 1 to $\boldsymbol{Z}$ and $\boldsymbol{P}$, respectively, to approximate the real function achieving ∞ by $p = 1, z = M$.

Algorithms are implemented in MATLAB 2014b [75] with optimization problems solved by Cplex 12.51 [46] on a platform with Intel(R) Xeon(R) CPU E5-1620 v2 @3.70 GHz. Simulations are run for carrying out the given ITT tasks using waterborne AGVs controlled by the proposed cost-effective RDMPC.

6.5.1 Safely fulfilling ITT tasks

Optimal trajectories of the three waterborne AGVs carrying out their assigned ITT tasks are shown in Figure 6.4a. Generally, all waterborne AGVs are able to track the shortest reference path between origin and destination pairs well. We observe two types of deviations along the paths. One is the relatively small deviations outside of conflict zone due to the existence of environmental uncertainties. The other is due to the encountering of waterborne AGVs in the conflict zone where deviations from the reference path are necessary to maintain a safety distance away from others. The trade-off among deviations, robustness level, and safety has been optimized since all online optimizations are successfully solved. Furthermore, waterborne AGVs arrive at their specified destinations punctually at 397s, 401s, and 401s, respectively, despite the possibly time consuming behaviors for collision avoidance and environmental uncertainties.

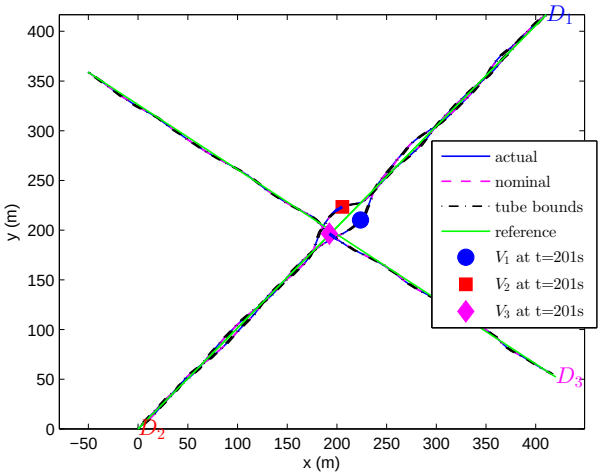
The proposed cost-effective RDMPC contains a bunch of uncertain trajectories in a tube that achieves both reference tracking and safety. The trajectory tubes of three waterborne AGVs are shown in Figure 6.4b which zooms in the conflict area of Figure 6.4a. Actual trajectories that are steered to nominal trajectories are guaranteed to stay within tubes if real uncertainties are within bounded sets. Note that although trajectory tubes overlay spatially, they do not overlay temporally at the same time. This is demonstrated by different positions of V_1, V_2, and V_3 at the a same time $t = 209$s plotted in different shapes as shown in Figure 6.4. Collisions are thus successfully avoided. Figure 6.5 further confirms overall safety by showing that Euclidean distances for all pairwise waterborne AGVs are above the safety distance line all the time. However, unlike deterministic scenarios where waterborne AGVs approach to each other until the minimal possible distance, i.e., the safety distance, is reached, the minimal distances in uncertain scenarios are a clear space above the safety distance line. The spaces intuitively act as buffers accounting for uncertainties as to be demonstrated later.

6.5.2 Convergence of distributed decision making

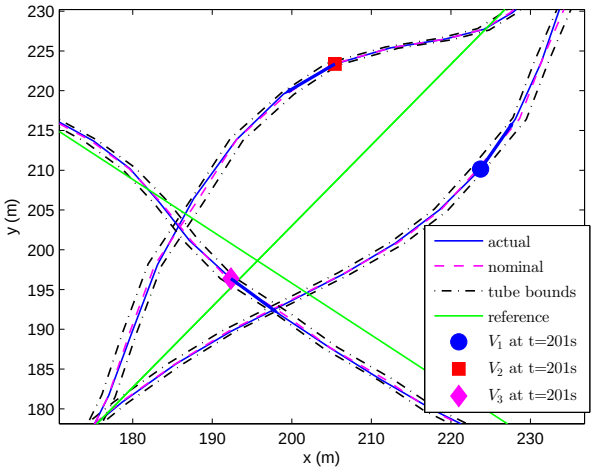
Waterborne AGVs with stochastic uncertainties compute in a distributed way and communicate to achieve overall satisfactions of coupling constraints and an overall cost-effective robustness level by Algorithms 6.1 and 6.3. The efficiency of the algorithms lies in the application of SOS1 branch and the integration of B&B and ADMM. Iterations with fractional solutions are terminated before convergence, and iterations with integer solutions follow ADMM stopping criteria (6.50) – (6.51). We show simulation results with integer solutions to demonstrate how feasibility and optimality convergence in a relaxed cost-effective RDMPC problem are achieved in this section, and illustrate the efficient B&B with SOS1 search procedure in Section 6.5.4.

At time step $k = 35$, three waterborne AGVs are coupled as graph $\mathcal{G}(k) = \{\mathcal{V}(k), \mathcal{E}(k)\}$ with $\mathcal{V}(k) = \{V_1, V_2, V_3\}$ and $\mathcal{E}(k) = \{(V_1, V_2), (V_1, V_3), (V_2, V_3)\}$. The second relaxed RDMPC problem, i.e., the second B&B node problem of this time step reaches integer solution agreement (Line 14 in Algorithm 6.3) in the first iteration. Therefore, the remaining iterations follow standard ADMM steps until convergence. The iterative satisfaction of coupling constraints is illustrated in Figure 6.6 which shows nominal distances over N_p between V_1 and V_2. The safety distance line defines the actual minimal separation distance between two waterborne AGVs. Nominal distances, however, are tightened due to the existence of uncertainties by appending a distance buffer as the red dashed lines shows to the actual safety distance. Waterborne AGVs with uncertainties controlled by the nominal control law are then still stay a safety distance away from each other, as demonstrated in Figure 6.6. During the first iterations, nominal distances at several prediction steps are below the distance buffer or even the safety distance; but by iterative communication and coordination, both V_1 and V_2 adjust their initial calculated trajectories which are driven above the distance buffer in the end. The distances for (V_1, V_3), and (V_2, V_3) show similar convergence behaviors.

Figure 6.7 shows objective convergence along with iterations. In Figure 6.7(a), the overall primal objective which is a sum of distributed calculated objectives from three waterborne AGVs as (6.37) converges to the centralized objective at the same time step. Figure 6.7(b) demonstrates that the intermediate objective J^j_{LP} defined in Section 6.4.2 converges to the optimal distributed objective J^*_{LP} from the below, which confirms the possibility of



(a) Trajectories of three waterborne AGVs.



(b) Trajectory tubes

Figure 6.4: Trajectories of waterborne AVGs for ITT

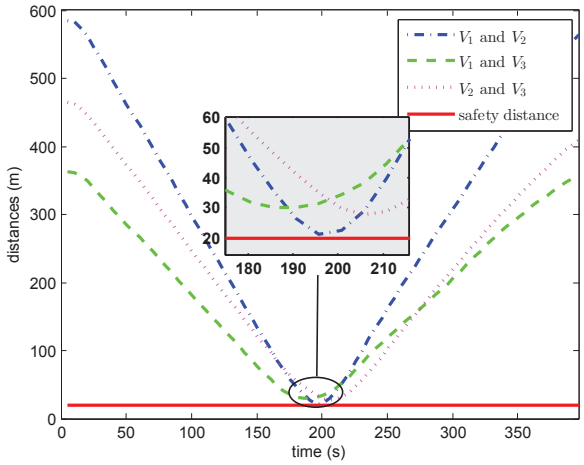


Figure 6.5: Distances between waterborne AGVs.

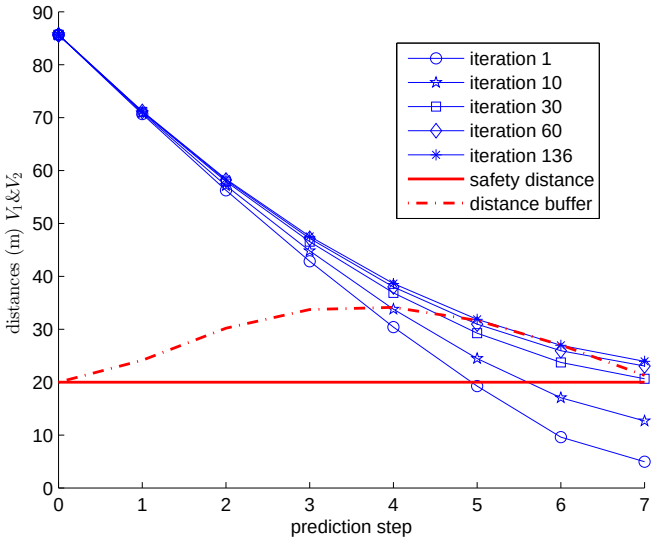


Figure 6.6: Robust distance iterations between waterborne AGVs 1 and 2 at time step $k = 35$.

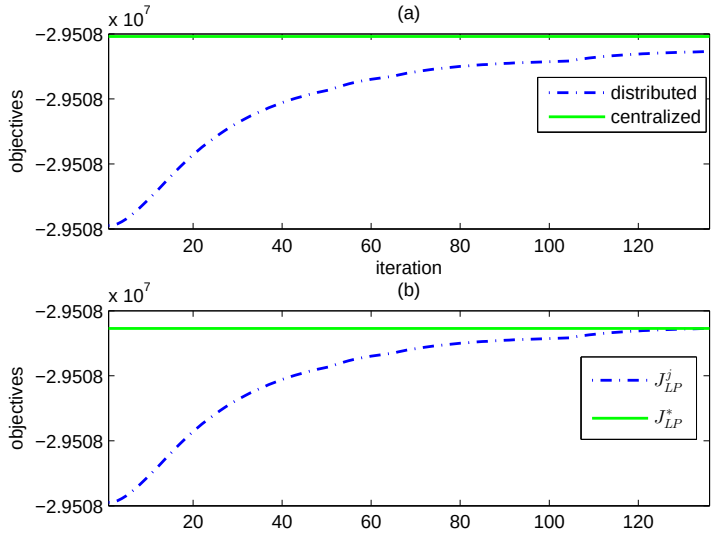


Figure 6.7: Objective iterations at time step $k = 35$.

early termination as in Proposition 6.2. Overall feasibility and optimality are further indicated by primal and dual residual convergence to specified tolerances in Figure 6.8. Primal residuals being small means that the consensus (6.42) is achieved and local solutions $\Delta\bar{\boldsymbol{r}}_p^j(k)$ are also satisfying coupling constraints (6.34). Dual residuals being small means that the iterative coordinator solutions do not change much any more. Overall safety and optimality are achieved when primal and dual residuals satisfy (6.50) and (6.51) at the final iteration 136.

6.5.3 Cost-effective robust tubes

This section explores the cost-effective robustness aspect of the proposed algorithm for cooperative distributed waterborne AGVs. The main idea is to explicitly consider system and uncertainty characteristics so that overall control performance including problem solvability and overall safety are guaranteed. This is achieved by using varying robustness levels with varying probabilities and uncertainty bounds as shown in Figure 6.9. Figure 6.9(a) shows the optimal robustness levels, i.e., probabilities of environmental disturbances V_1 can be robust to. The deviations from 100% are necessary for the feasibility of online optimizations and deviations are as small as possible. All robustness levels are higher than 95% to ensure high safety level. Note that when waterborne AGVs encounter in close proximity between 100s – 300s, robustness levels slightly drop to account for possible large tracking errors due to collision avoidance. This also showcases the optimal trade-off property of the proposed algorithm. Subplot (b) shows the uncertainty bounds corresponding to the robustness levels and uncertainties that actually impose on waterborne AGVs. Real uncertainties are all contained in the optimally bounded sets, which demonstrates safety of high robustness levels. Those extreme uncertainties beyond the bounds, though possible, are very rare

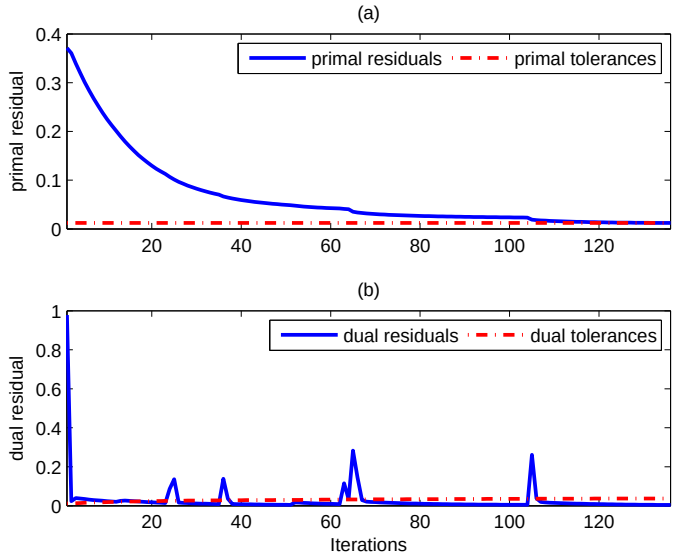


Figure 6.8: Primal and dual residuals at time step $k = 35$.

to occur. Robustness against them could degrade system performance dramatically and is thus deemed as not cost-effective. Throughout the simulation, infeasibility is not observed. This might be because a large magnitude of the uncertainty has been taken care of by the designed robustness. When uncertainties that do cause infeasibility issues in cost-effective RDMPC, rarely though, safety/recovery procedures need to be activated.

Cost-effective robust satisfaction of coupling constraints has been shown in Figure 6.6 with nominal distances tightened by an extra distance buffer. Cost-effective robust satisfactions of local constraints on speed and control inputs are shown in Figure 6.10 and Figure 6.11, respectively. The common patterns for Figure 6.10 and Figure 6.11 are that the actual speed and control input trajectories are both well constrained within system limits denoted by the red lines and are both a clear distance away from their corresponding limits. This is the saved space for uncertainties, i.e., the price of robustness. Figure 6.10 and Figure 6.11 differ in that the actual and nominal trajectories for control input coincide while for speed not. This is because MPC applies, in a calculated optimal sequence, only the first control input which by (6.6) and (6.9) has no uncertainty involved. However, due to the introduction of uncertainties via (6.9), predicted control inputs over $1,2,...,N_p-1$ are necessarily tightened. Figure 6.12 shows the nominal control input trajectories and tubes over N_p at one time step. The zero initial control input tightening offsets can also be clearly seen in Figure 6.12. Similarly, Figure 6.13 shows the speed tubes over N_p. Speed and control input tubes evolve according to (6.14) and (6.15), respectively. The speed tube sizes approximately increase over N_p due to set additions narrowing the feasible region for nominal speeds. Therefore, nominal speeds decrease as the prediction step proceeds. However, online optimization problems remain feasible due to the explicit integration of the uncertainty bounds. Tightly fit tubes in Figure 6.10, Figure 6.13, and Figure 6.12 but still with a feasible nominal solution illustrate this.

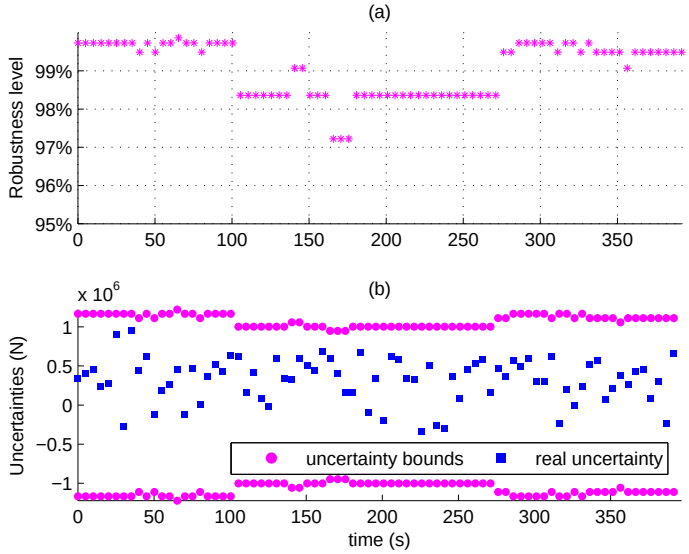


Figure 6.9: Varying robustness level: probability and uncertainty bounds.

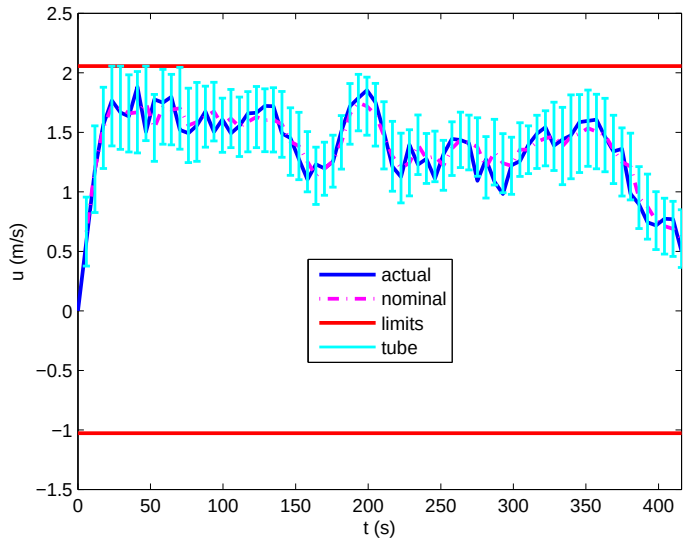


Figure 6.10: Speed trajectories with tubes of the whole simulation.

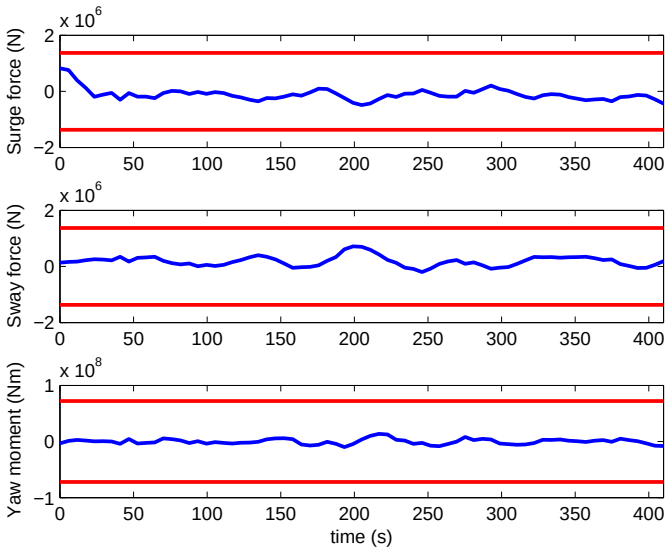


Figure 6.11: Control inputs of the whole simulation.

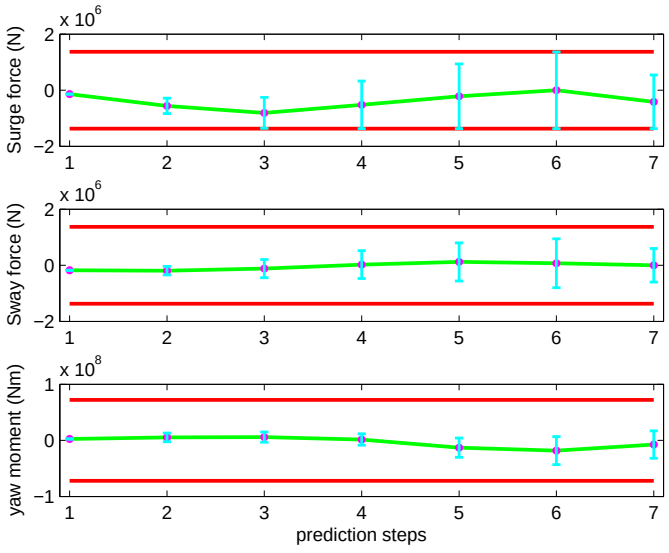


Figure 6.12: Predicted control inputs with tubes at one time step.

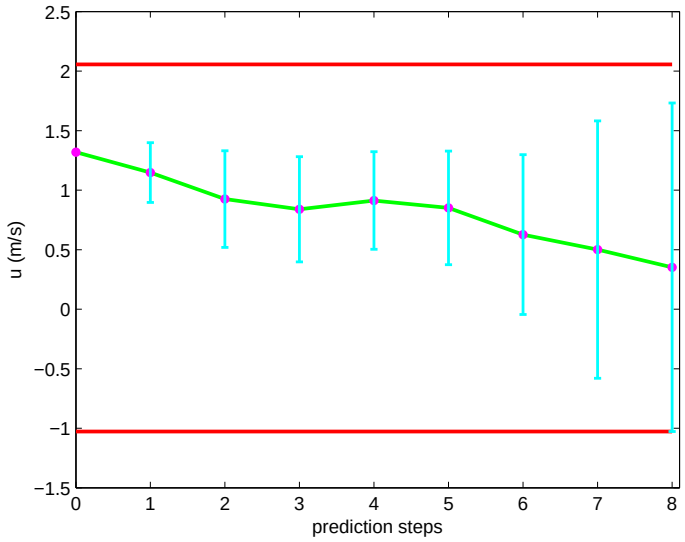


Figure 6.13: Predicted speed with tubes at one time step.

6.5.4 Efficient B&B search

Efficient B&B search retrieves exact solutions from relaxed RDMPC problems by exploiting the special structure of SOS1 constraints (6.35) and the intermediate results of ADMM. With the proposed cost-effective RDMPC algorithm, we observe at most two branching operations throughout the simulation. Figure 6.14 and Figure 6.15 show the B&B search trees with height one and two, respectively. In both figures, the tree nodes are attached with SOS1 constraints on subset of variables and the upper (U) and lower (L) bounds by solving this node problem; the numbers left to the nodes indicate the sequence by which node problems are dealt with. Note that the sum of the subset of variables in SOS1 constraints should equal one, e.g., $\sum_{i=1}^{21} a_i = 1$ for the top node in Figure 6.14, but $\sum_{i=1}^{21} a_i$ has been used for simplicity. The division of the variable set at a parent node to the sets at two child nodes are based on fractional solutions from the parent node problem by deciding the branching point as (6.53) – (6.54). The upper bound, i.e., incumbent objective, is either updated with the better bound produced by the current problem with integer solutions, or otherwise inherited from the parent node. The lower bound is either the objective solving the relaxed cost-effective RDMPC problem, or ∞ if the relaxed problem is infeasible.

Overall, the algorithm first searches a feasible solution, and then either improve this solution or verify the optimality of this solution by implicitly enumerating all possible solutions. E.g., for the tree in Figure 6.15, node 1 solves the relaxed *Problem 3* and the fractional solutions after 20 iterations suggest a branching point $i^* = 15$. Therefore, $\sum_{i=1}^{14} a_i = 1$ and $\sum_{i=15}^{21} a_i = 1$ are added to the two child nodes, node 2 and node 5, respectively. In a similar way as node 1, node 2 generates two child nodes of which node 3 has an integer feasible solution. Since the optimal objective of node 3 is smaller than the incumbent objective, i.e., its parent upper bound ∞, the incumbent objective and incumbent solution are updated.

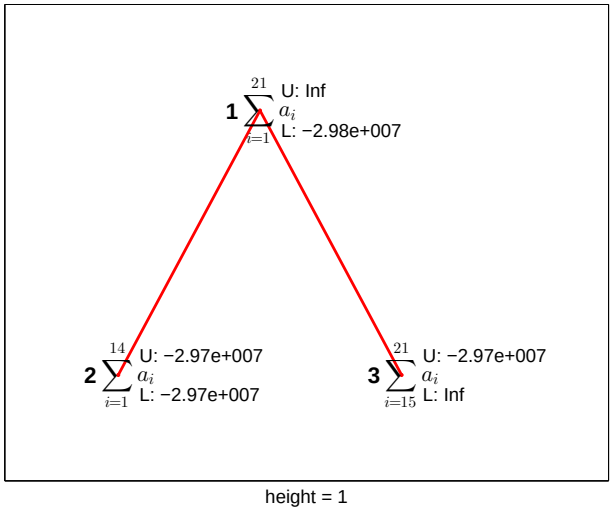


Figure 6.14: B&B search tree with height 1.

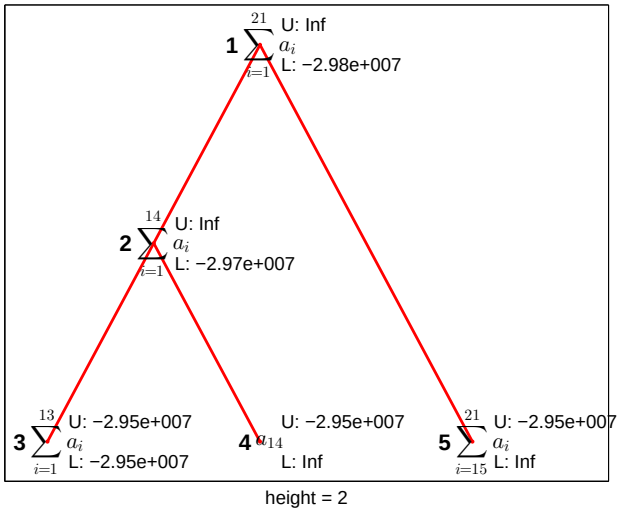


Figure 6.15: B&B search tree with height 2.

Problems at node 4 and node 5 are infeasible and are pruned immediately. Therefore, generally only one node needs to iterate to convergence in a search tree and thus making the algorithm efficient.

6.6 Conclusions

In this chapter, we have proposed a cost-effective robust distributed MPC (RDMPC) control approach for multiple waterborne AGVs that carry out ITT tasks in scenarios with uncertainties. Environmental disturbances due to wind, waves, and currents are assumed to be predicted by the port authority. However, there still exist stochastic uncertainties in the predictions, as modeled in Chapter 3. Each waterborne AGV computes locally those actions that fulfill its assigned ITT task by tracking a reference path with arrival time awareness in an energy-efficient way with a cost-effective price of being robust. In our simulations, all robustness levels are higher than 95% to ensure high safety level. Moreover, overall minimal cost, robustness level, and coupling collision avoidance constraints have been satisfied by communicating within a neighborhood. The cost-effective robustness extends the tube-based robust approach for bounded uncertainties to stochastic uncertainties with infinite support by explicitly considering system and uncertainty characteristics. The cost-effective RDMPC problem for coupled waterborne AGVs has been solved exactly and efficiently by integrating a special type of B&B and ADMM. With the proposed efficient searching strategy, at most two branching operations at each time step are required throughout the simulation. Simulation results of an ITT case study in the port of Rotterdam illustrate that the proposed cost-effective RDMPC algorithm is effective for controlling multiple waterborne AGVs with uncertainties in carrying out ITT tasks, which answers the fourth Key Research Question in Chapter 1.

So far, control problems for waterborne AGVs carrying out ITT tasks have been considered in Chapters 4 – 6. In all these problems, schedules regarding the assignment of ITT tasks to waterborne AGVs and the routing of waterborne AGVs are assumed given. In the next chapter, coordination considering both the control and a higher level scheduling problems of waterborne AGVs will be discussed.

Chapter 7

Closed-loop scheduling and control for autonomous Inter Terminal Transport

In Chapters 4 – 6, effective controllers are developed given one ITT request per waterborne AGV. The scheduling level decisions are assumed made and passed to the controllers acting as the references that waterborne AGVs should track. The scheduling and control problems are solved in an open-loop way. This chapter considers the scheduling of ITT tasks for multiple waterborne AGVs and proposes a closed-loop energy-efficient scheduling and control framework to realize an autonomous waterborne ITT system. The scheduling problem is formulated as a pick-up and delivery problem (PDP) that considers speed optimization and safe intervals between berthing time slots of different waterborne AGVs at the same berth. Waterborne AGVs are controlled in a cooperative distributed way to fulfill assigned schedules as in Chapter 5.

The research discussed in this chapter has been published in [153].

7.1 Introduction

In the literature, multi-vehicle scheduling problems are largely formulated as Vehicle Routing Problems (VRPs) [130] determining the assignment of vehicles to transport tasks and the sequence of points possibly with temporal requirements that a vehicle should visit. However, assumptions are implicitly made that vehicles are dimensionless mass points finishing assigned tasks as scheduled perfectly without consideration safety intervals between vehicles. Moreover, the scheduling and control level problems are typically solved separately, as illustrated in Figure 1.2. Sharing the common aims of making economical and environmentally friendly decisions, scheduling and controlling waterborne AGVs given multiple ITT requests are expected to achieve further benefits via a tighter interaction of these two.

In this chapter, we propose a closed-loop scheduling and control approach for a fleet of waterborne AGVs to realize an autonomous ITT system. Closed-loop means that both scheduling and control levels make decisions online based on system states measured at a

fast sampling rate. Decisions are still made hierarchically to guarantee tractability. Moreover, a new PDP scheduling model considering necessary time intervals between different waterborne AGVs visiting a particular berth is proposed. Furthermore, we propose a partial scheduling problem that is efficient to solve, and an interaction model that integrates the scheduling and control problems. Solving the scheduling problem generates for each waterborne AGV a sequence of terminals to visit to load or unload certain amount of containers sailing at an energy optimal speed arriving/departing at coordinated berthing times, while still satisfying service time windows. Cooperative distributed model predictive control based on the fast ADMM algorithm of Chapter 5 is then adopted by the group of involved waterborne AGVs to accomplish the schedules safely and accurately. The main advantage of using a closed-loop scheme over an open-loop scheme is that real-time factors such as unconsidered physical system limits, disturbances, and collision avoidance that are difficult, if not impossible, to be integrated in a scheduling problem can be reflected timely by the online updated schedules.

The remainder of this chapter is organized as follows. The overall problem statement for an autonomous ITT system using waterborne AGVs is first introduced in Section 7.2. Then in Section 7.3, the energy efficient scheduling problem with coordinated berthing times is formulated. The closed-loop scheduling and control based on a real-time coupling speed assignment problem and an interaction model are proposed in Section 7.4. In Section 7.5, simulation experiments and results are presented, followed by conclusions of this chapter in Section 7.6.

7.2 Problem statement

We consider an autonomous ITT system: a fleet of waterborne AGVs that handles a set of emerging ITT requests to transport specified amounts of containers between specified origins and destinations within specified time windows autonomously in an energy efficient way. Without loss of generality, two simplifications are made: *1*) Each terminal has one waterborne AGV berth. In practice, a terminal can have multiple berths, which can, however, be viewed as multiple pickup/delivery locations. Therefore, the problem formulation is essentially not changed for one berth per terminal; and *2*) ITT requests are decoupled between different planning horizons so that requests arising within each planning horizon are completed within that horizon.

For the scheduling problem, each waterborne AGV has a finite capacity which can accommodate mixing containers from different requests. Besides, waterborne AGVs use "environmentally friendly" engines and perform "slow steaming" by cruising at lower speeds if possible. There is no central depot for waterborne AGVs and they stay at the park lot of the final service berth. Finally, waterborne AGVs are with certain dimensions and need to keep a certain safety distance from others to avoid collisions. Collision avoidance is achieved among moving waterborne AGVs by cooperative distributed control as presented in Chapter 5 while waterborne AGVs visiting the same berth to perform loading or unloading operations are yet to be coordinated. Figure 7.1 shows an illustration of an ITT system with a fleet of three waterborne AGVs, six waterborne AGV berths and 12 routes[1].

[1]These routes are all with much shorter distances by water than by land while routes connecting, e.g., berth 5 and 6, with short land distances are considered by other ITT modes.

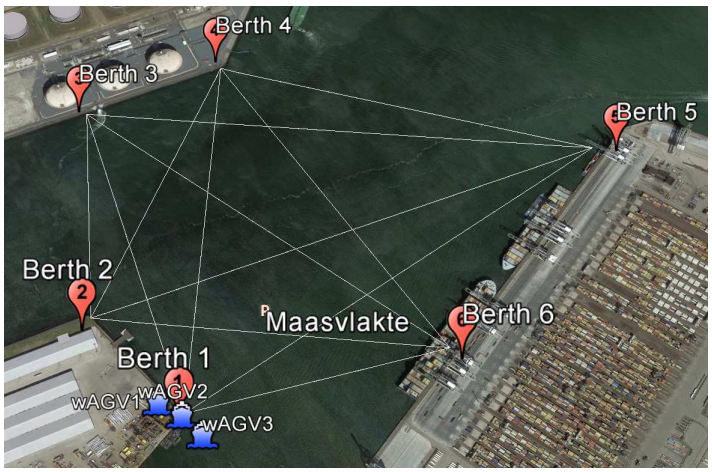


Figure 7.1: Illustration of an ITT system with six berths for three waterborne AGVs.

In light of the above ITT network and available fleet of waterborne AGVs, the list of ITT requests should be available upon making decisions. In particular, each ITT request is associated with information on seven aspects:

1. Request ID which is sorted by all requests' release times;
2. Origin berth ID corresponding to the pick-up location;
3. Destination berth ID corresponding to the delivery location;
4. Release time defining when a set of containers are ready to be shipped, being the earliest time that the loading service can start;
5. Due time defining when the set of containers are ready for subsequent operations, being the latest time of completing this request including the unloading time at the destination berth; and
6. Volume of the set of containers to be shipped in TEUs;
7. Service time for loading/unloading the set of containers.

In addition, since delays or waiting times do occur in reality and meeting hard time windows may fail in finding a feasible solution, requests are allowed to be serviced within soft time windows, but customer inconvenience cost will incur if not within hard time windows. Note that trade-offs can also be made by using more waterborne AGVs to reduce delays. Containers with the same request ID cannot be shipped by different waterborne AGVs as assumed in Chapter 1. Finally, containers have to be transported without transhipment, i.e., loading and unloading operations happen exactly once for each request.

The autonomous ITT system runs in a closed-loop fashion, i.e., both scheduling and control level problems in Figure 1.2 are solved in real-time using updated system states. Control of a fleet of waterborne AGVs is realized as in Chapter 5 while the energy efficient scheduling problem as well as the closed-loop scheduling and control design are presented in the following two sections.

7.3 Energy efficient scheduling of ITT using waterborne AGVs

Traditionally, the scheduling problem of a fleet of ships traveling back and forth among terminals to transport goods relies on human dispatchers necessarily with high competence and experience. Complex decisions need to be made satisfying various possibly conflicting objectives (e.g., saving energy by sailing at low speeds while meeting time windows at high speeds) considering transport requests and available vehicle lists. Assignment of waterborne AGVs to routes and timing can break such an operator. In face of real-time operational delays and uncertainties, the problem can frustrate human dispatchers even more. We next present a scheduling model based on mixed integer programming for ITT using waterborne AGVs to ease the workload of human dispatchers. We first introduce relevant notations including input parameters to the model and decision variables to be solved from this model. Then the mathematical model is presented which is further transformed into a mixed integer linear programming (MILP) problem to reduce required computation times.

7.3.1 Notations

The planning horizon considered within which a set of ITT requests $\mathcal{R}$ among the set of berths $\mathcal{B}$ arise is $\left[0, T_{\mathrm{p}}\right]$. For each request $i \in \mathcal{R}$, we denote a 7-element tuple $\langle i, p_i, d_i, t_{i,\min}, t_{i,\max}, q_i, s_i \rangle$ to represent the information associated with request i as described in Section 7.2, i.e., request ID, pick-up berth, delivery berth, release time, due time, volume, and service time. For each pick-up location p_i, a positive load $+q_i$ is attached, and each delivery location d_i, a negative load $-q_i$ attached. The set of n_v waterborne AGVs is $\mathcal{V}$ and homogenous. The set for start locations for all waterborne AGVs is defined as $\mathcal{V}_{\mathrm{o}} = \{1, ..., n_v\}$ and the set for end locations as $\mathcal{V}_{\mathrm{e}} = \{n_v + 2n + 1, ..., 2n_v + 2n\}$ with $n = |\mathcal{R}|$. All waterborne AGVs have the same capacity Q in TEUs, curb weight m and cruising speed range $[u_{\min}, u_{\max}]$. All TEU of containers are assumed to have the same weight m_c.

As has been discussed before, ITT scenarios inevitably involve waterborne AGVs shuttling back and forth, and thus pick-up and delivery locations of different requests might actually be the same physical berths. This is one of the main differences of our scheduling problem with land-based VRPs [130] or PDPs [115] based on assumptions of distinct pick-up and delivery locations and vehicles visit each location exactly once. We, hence, define virtual pick-up and delivery node sets $\mathcal{P}_{\mathrm{n}} = \{n_v + 1, n_v + 2, ..., n_v + n\}$ and $\mathcal{D}_{\mathrm{n}} = \{n_v + n + 1, n_v + n + 2, ..., n_v + 2n\}$, respectively. Then, our scheduling problem is defined over the virtual graph $\mathcal{G}_s = (\mathcal{N}, \mathcal{A})$ with node set $\mathcal{N} = \mathcal{P}_{\mathrm{n}} \cup \mathcal{D}_{\mathrm{n}} \cup \mathcal{V}_{\mathrm{o}} \cup \mathcal{V}_{\mathrm{e}}$ and arc set $\mathcal{A} = \big\{(i,j) \big| (i,j) \in \big((\mathcal{P}_{\mathrm{n}} \cup \mathcal{D}_{\mathrm{n}}) \times ((\mathcal{P}_{\mathrm{n}} \cup \mathcal{D}_{\mathrm{n}})) \cup \big\{(i,j) \big| i \in \mathcal{V}_{\mathrm{o}}, j \in \mathcal{P}_{\mathrm{n}} \cup \mathcal{D}_{\mathrm{n}} \big\} \cup \{(i,j) \,|\, i \in \mathcal{P}_{\mathrm{n}} \cup \mathcal{D}_{\mathrm{n}}, j \in \mathcal{V}_{\mathrm{e}} \big\}, i \neq j \big\}$. The physical locations of nodes in virtual graph $\mathcal{G}_s$ are mapped as a vector L corresponding to $\mathcal{N}$. Denote d_{ij} as the travel distance between nodes i and j for all $(i,j) \in \mathcal{A}$. Note that since waterborne AGVs stay at their final service berth, the locations for virtual end nodes $\mathcal{V}_{\mathrm{o}}$ vanish and distance $d_{ij} = 0$ if $i \in \mathcal{P}_{\mathrm{n}} \cup \mathcal{D}_{\mathrm{n}} \cup \mathcal{V}_{\mathrm{o}}, j \in \mathcal{V}_{\mathrm{e}}$. For duplicated elements in L (same berths), we further cluster the corresponding nodes as set $\mathcal{C}_b = \{i \,|\, L_i = b, b \in \mathcal{B}\}$. For the nodes in a same set $\mathcal{C}_b$, if they are visited by different waterborne AGVs, a time interval T is imposed to the service time slots of the waterborne AGVs to keep safety considering waterborne AGV dimensions.

The following decision variables are introduced to solve the scheduling problem:

- Binary variables: x_{ijv} for $(i,j) \in \mathcal{A}$ and $v \in \mathcal{V}$ equals to 1 if waterborne AGV v travels from node $i \to j$ and 0 otherwise;
- Binary variables: z_{iv} for $i \in \mathcal{N}$ and $v \in \mathcal{V}$ equals to 1 if node i is visited by waterborne AGV v and 0 otherwise;
- Binary variables: I_{ij} for $i,j \in \mathcal{C}_b, b \in \mathcal{B}, i \neq j$ equals to 1 if node i is visited before node j and 0 otherwise;
- Binary variables: S_{ij} for $i,j \in \mathcal{C}_b, b \in \mathcal{B}$ equals to 1 if nodes i,j are visited by different waterborne AGVs and 0 otherwise;
- Integer variables: y_i for $i \in \mathcal{N}$ denotes the load on board the waterborne AGV upon arriving node i;
- Continuous variables: a_i for $i \in \mathcal{N}$ specifies the arrival time at node i;
- Continuous variables: w_i for $i \in \mathcal{N}$ is the waiting time at node i;
- Continuous variables: d_i for $i \in \mathcal{N}$ is the delay time at node i;
- Continuous variables: u_{ij} for $(i,j) \in \mathcal{A}$ is the speed a waterborne AGV travels at on leg $i \to j$.

Additional auxiliary variables for the transformation into an MILP problem will be introduced in Section 7.3.3.

7.3.2 Mixed integer programming problem

The overall goal is to compute a set of schedules that minimize the cost of fulfilling all requests in $\mathcal{R}$ according to some cost metrics while satisfying various constraints. For our case, a mixed integer programming problem is formulated as follows:

$$\begin{aligned}\min\ & c_1 \sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{P}_\mathrm{n}\cup\mathcal{D}_\mathrm{n}} \sum_{i\in\mathcal{V}_\mathrm{o}} x_{ijv} + c_2 \sum_{v\in\mathcal{V}} \sum_{(i,j)\in\mathcal{A}} x_{ijv}(m_c y_j + m) d_{ij} + c_3 \sum_{(i,j)\in\mathcal{A}} u_{ij}^2 d_{ij} \\ & + c_4 \left\| A_{\mathcal{V}_e} - A_{\mathcal{V}_o} \right\|_1 + c_5 \|\mathbf{w}\|_1 + c_6 \|\mathbf{d}\|_1 \end{aligned} \tag{7.1}$$

subject to

$$\sum_{v\in\mathcal{V}} z_{iv} = 1 \qquad \forall i \in \mathcal{N}, \tag{7.2}$$

$$z_{iv} = z_{(i+n_r)v}, \qquad \forall i \in \mathcal{P}_\mathrm{n}, v \in \mathcal{V}, \tag{7.3}$$

$$z_{ii} = 1, \qquad \forall i \in \mathcal{V}_\mathrm{o}, \tag{7.4}$$

$$\sum_{j\in\mathcal{N}} x_{ijv} = \sum_{j\in\mathcal{N}} x_{jiv} = z_{iv}, \qquad \forall i \in \mathcal{N}, v \in \mathcal{V}, \tag{7.5}$$

$$\sum_{j\in\mathcal{N}/\mathcal{V}_\mathrm{e}} x_{v_\mathrm{o} jv} = 1, \qquad \forall v \in \mathcal{V}, \tag{7.6}$$

$$\sum_{i\in\mathcal{N}/\mathcal{V}_\mathrm{o}} x_{iv_\mathrm{d} v} = 1, \qquad \forall v \in \mathcal{V}, \tag{7.7}$$

$$a_i \leqslant A_{i+n_r}, \qquad \forall i \in \mathcal{P}_\mathrm{n}, \tag{7.8}$$

$$x_{ijv} = 1 \Rightarrow \max(A_i, t_{i,\min}) + s_i + d_{ij}/u_{ij} = A_j, \qquad \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \tag{7.9}$$

$$t_{i,\min} - w_i \leqslant a_i \leqslant t_{i,\max} - s_i + d_i, \qquad \forall i \in \mathcal{N}, \tag{7.10}$$

$$0 \leqslant w_i \leqslant w_{\max}, \qquad \forall i \in \mathcal{N}, \tag{7.11}$$

$$0 \leqslant d_i \leqslant d_{\max}, \qquad \forall i \in \mathcal{N}, \tag{7.12}$$

$$I_{ij} + I_{ji} = 1, \qquad \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, \tag{7.13}$$

$$S_{ij} = 1 - \sum_{v \in \mathcal{V}} z_{iv} z_{jv}, \qquad \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, \tag{7.14}$$

$$I_{ij} S_{ij} = 1 \Rightarrow \max(a_i, t_{i,\min}) + s_i + T \leqslant A_j, \qquad \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, \tag{7.15}$$

$$y_{v_o} = y_{v_e} = 0, \qquad \forall v \in \mathcal{V}, \tag{7.16}$$

$$x_{ijv} = 1 \Rightarrow y_i + q_i = y_j, \qquad \forall i \in \mathcal{N}, v \in \mathcal{V}, \tag{7.17}$$

$$0 \leqslant y_i \leqslant Q, \qquad \forall i \in \mathcal{N}, \tag{7.18}$$

$$u_{\min} \leqslant u_{ij} \leqslant u_{\max}, \qquad \forall (i,j) \in \mathcal{A}, \tag{7.19}$$

$$x_{ijv}, z_{iv}, I_{ij}, S_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \tag{7.20}$$

where the objective (7.1) contains six cost terms that are related to energy efficient schedules for waterborne AGVs. The first term counts the number of waterborne AGVs deployed for the set of requests $\mathcal{R}$. The fleet of deployed waterborne AGVs is not necessarily the same with the fleet of available waterborne AGVs; we always minimize the number of deployed waterborne AGVs considering high fixed deployment cost. Both the second and third terms measure the cost of energy consumption and emissions traveling from node $i \to j$. The pollution-routing problem [7] employed similar emission measurement terms. Cost term 2 is incurred due to the weight including waterborne AGV curb weight and the weight of containers on board of the waterborne AGV. Cost term 3 reflects the nonlinear dependence of energy consumption on cruising speed and distance. "Slow steaming" is imposed by minimizing this term if possible. The fourth term considers the total sojourn time of all waterborne AGVs. Departure times from starting locations are also optimized with this formulation. The last two terms account for customer inconvenience measured by waiting and delay times, respectively. The trade-off among these cost penalties is balanced by weight parameters $c_1, c_2, \cdots, c_6$.

Constraint (7.2) represents that each node is visited exactly by one waterborne AGV. By constraints (7.3) and (7.4), we ensure that pick-up and delivery nodes of a particular request are visited by the same waterborne AGV and all waterborne AGVs visit their own starting nodes, respectively. Constraint (7.5) restricts that a waterborne AGV only enters and leaves a node if it visits that node. Constraints (7.6) and (7.7) impose that each waterborne AGV starts and ends at the right locations, respectively. Constraints (7.8) – (7.15) together impose time constraints. Specifically, inequality (7.8) guarantees that pick-up nodes are visited before delivery nodes. Constraint (7.9) enforces time consistency where the *max* operation indicates that loading/unloading services cannot start earlier than the release times of requests. Time window constraints are specified by (7.10) - (7.12). The coordinated berthing times taking waterborne AGV dimensions and safety distances into consideration are realized with constraints (7.13) – (7.15). The logic in (7.15) implies that if node i, j relate to the same physical location ($i, j \in \mathcal{C}_b$) and are visited by different waterborne AGVs ($S_{ij} = 1$) and node i is visited before node j ($I_{ij} = 1$), that then the arrival time of the waterborne AGV behind should be later than the departure time of the earlier waterborne AGV at least for a time T. This is a novel feature of our waterborne AGV scheduling problem. VRPs and variants have typically assume vehicles as dimensionless mass points without consideration of safety distances. Load consistence and capacity constraints are introduced via (7.16) – (7.18). Lastly, cruising speed is bounded by (7.19), and (7.20) defines binary variables.

The above mixed integer programming problem (7.1) – (7.20) involves several nonlinearities:

- the multiplication of binary variable x_{ijv} and integer load variable y_j in the second cost term of (7.1);
- the quadratic energy function of speed variable u_{ij} in the third cost term of (7.1);
- the reciprocal of speed u_{ij} in (7.9);
- logic implications in constraints (7.9), (7.15) and (7.17); and
- the multiplications of binary variables $z_{iv}z_{jv}$ in (7.14) and $I_{ij}S_{ij}$ in (7.15).

All these nonlinearities bring about even more challenges to finding an optimal solution to the already notorious NP-hard routing problem. We next present transformations of these nonlinearities to obtain an easier to solve MILP problem.

7.3.3 Transformations into linearity

Linearizations of the above reported nonlinearities rely mainly on two techniques: discretization [7, 25], and logic and integer formulations [140]. We first deal with the nonlinearities caused by nonlinear functions of speed by discretization.

Generally, two discretization approaches are proposed: discrete speeds as in [7] and discrete travel times as in [25]. Essentially, these two approaches are the same since speeds and travel times are related by a constant travel distance. We apply discrete speeds as in [7] due to its more intuitive formulation in modeling the "slow steaming" effect. The continuous cruising speed range $[u_{\min}, u_{\max}]$ is discretized by equal intervals $(u_{\max} - u_{\min})/n_u$ into a set of n_u speed levels $[u_{r,\min}, u_{r,\max}]$ for $r = 1, ..., n_u$. An average speed is then calculated as $\bar{u}_r = (u_{r,\min} + u_{r,\max})/2$ and assigned to that level. Therefore, the speed optimization in the continuous range $[u_{\min}, u_{\max}]$ becomes the optimal speed selection in the discrete speed set $\{\bar{u}_r \,|r = 1, ..., n_u\}$. Correspondingly, we introduce binary variables b_{ijrv} equal to 1 if waterborne AGV v travels from node $i \to j$ at speed $\bar{u}_r$. Then, the third cost term, which is a quadratic speed function, is rewritten as:

$$\sum_{(i,j)\in\mathcal{A}} u_{ij}^2 d_{ij} = \sum_{(i,j)\in\mathcal{A}} \left(\sum_{r=1}^{n_u} \bar{u}_r^2 b_{ijrv} \right) d_{ij}. \tag{7.21}$$

Similarly, the reciprocal speed term in (7.9) is rewritten as:

$$d_{ij}/u_{ij} = \sum_{r=1}^{n_u} (d_{ij}/\bar{u}_r) b_{ijrv}. \tag{7.22}$$

Note that different from the VRP in [7] with distinct visiting locations, we have duplicated pick-up and delivery locations, which leads to $d_{ij} = 0$ when nodes i, j are actually the same physical berths. Therefore, the relation between b_{ijvv} and x_{ijv} is constrained as:

$$\sum_{r=1}^{n_u} b_{ijrv} = \delta_{ij} x_{ijv}, \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \tag{7.23}$$

where δ_{ij} is a binary constant equal to 1 if $d_{ij} > 0$ and 0 if d_{ij}. This formulation enforces a zero speed on arc (i, j) if nodes i, j are the same physical berths and a non-zero speed otherwise.

To deal with the nonlinearities due to multiplications of binary and integer variables $x_{ijv}y_j$ in (7.1), multiplications of binary and binary variables $z_{iv}z_{jv}$ and $I_{ij}S_{ij}$ in (7.14) and (7.15), respectively, and logic implications in (7.9), (7.15) and (7.17), the following linearizing approaches based on logic and integer formulations [140] are implemented.

Introduce auxiliary real variables $X_{ijv} = x_{ijv}y_j$, then we are able to replace the nonlinear term in $x_{ijv}y_j$ in (7.1) by X_{ijv} subject to the following set of linear constraints:

$$\begin{aligned} X_{ijv} &\leqslant Q x_{ijv}, && \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \\ X_{ijv} &\geqslant 0, && \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \\ X_{ijv} &\leqslant y_j, && \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \\ X_{ijv} &\geqslant y_j - Q(1 - x_{ijv}) && \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}. \end{aligned} \tag{7.24}$$

The equivalence is due to the bound on variables y_j as constraints (7.18).

Slightly differently, we replace the nonlinear terms in (7.14) and (7.15) with auxiliary binary variables $Z_{ijv} = z_{iv}z_{jv}$ and $Y_{ij} = I_{ij}S_{ij}$ along with the following two sets of linear constraints, respectively:

$$\begin{aligned} -z_{iv} + Z_{ijv} &\leqslant 0, && \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, v \in \mathcal{V}, \\ -z_{jv} + Z_{ijv} &\leqslant 0, && \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, v \in \mathcal{V}, \\ z_{iv} + z_{jv} - Z_{ijv} &\leqslant 1, && \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, v \in \mathcal{V}, \end{aligned} \tag{7.25}$$

$$\begin{aligned} -I_{ij} + Y_{ij} &\leqslant 0, && \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, \\ -S_{ij} + Y_{ij} &\leqslant 0, && \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, \\ I_{ij} + S_{ij} - Y_{ij} &\leqslant 1, && \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}. \end{aligned} \tag{7.26}$$

Finally, logic implications in (7.9) are transformed as:

$$\begin{aligned} \max(a_i, t_{i,\min}) + s_i + \sum_{r=1}^{n_u} (d_{ij}/\bar{u}_r) b_{ijrv} &\leqslant A_j + M_{ij}^1 (1 - x_{ijv}), && \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \\ -\left(\max(a_i, t_{i,\min}) + s_i + \sum_{r=1}^{n_u} (d_{ij}/u_r) b_{ijrv} \right) &\leqslant -A_j - m_{ij}^1 (1 - x_{ijv}), && \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \end{aligned} \tag{7.27}$$

with $M_{ij}^1 = t_{i,\max} + w_{\max} + s_i + d_{ij}/u_{\min} - (t_{i,\min} - w_{\max}), m_{ij}^1 = t_{i,\min} + s_i - (t_{i,\max} + w_{\max})$, and (7.17) as:

$$\begin{aligned} y_i + q_i &\leqslant y_j + M_{ij}^2 (1 - x_{ijv}), && \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \\ -(y_i + q_i) &\leqslant -y_j - m_{ij}^2 (1 - x_{ijv}), && \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \end{aligned} \tag{7.28}$$

with $M_{ij}^2 = Q + q_i, m_{ij}^2 = q_i - Q$, and (7.15) simply as:

$$\max(a_i, t_{i,\min}) + s_i + T \leqslant A_j + M_{ij}^3 (1 - Y_{ij}), \qquad \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, , \tag{7.29}$$

with $M_{ij}^3 = t_{j,\max} + w_{\max} + s_i + T - (t_i - w_{\max})$.

So far, we have transformed the nonlinear mixed integer programming problem (7.1) – (7.20) into an MILP problem by replacing nonlinear terms in the cost function and constraints with auxiliary variables and linear constraints as formulated as (7.21) - (7.29). Schedules generated by solving the MILP problem are, for each waterborne AGV $v \in \mathcal{V}$, sequences of nodes $\mathcal{N}_v = \{i \,|\, i \in \mathcal{N}, z_{iv} = 1, v \in \mathcal{V}\}$ to visit, the corresponding arrival times $\mathcal{A}_v = \{a_i \,|\, z_{iv} = 1, v \in \mathcal{V}\}$ in ascending order, load/unloading volumes $Q_v = \{q_i \,|\, z_{iv} = 1, v \in \mathcal{V}\}$, as well as traveling speeds $\mathcal{U}_v = \{u_{ij} \,|\, x_{ijv} = 1, (i,j) \in \mathcal{A}, v \in \mathcal{V}\}$ on each leg.

7.4 Real-time closed-loop scheduling and control

Ideally, a seamless integration of scheduling and control problem requires that all decisions are made simultaneously achieving objectives and satisfying various constraints at both levels, which still remains an open, though important, issue. The main technical challenges lie in the different time horizons and different nature of decisions (continuous time and discrete events) which result in a highly complex problem to solve considering current computing power. A descriptive integrated problem **P** could be defined as:

$$\begin{aligned} (\mathbf{u}_\mathrm{s}^*, \mathbf{u}_\mathrm{c}^*) := & \arg\min J(J_\mathrm{s}, J_\mathrm{c}) \\ \text{Subject to} \quad & \text{Scheduling constraints } (7.2) - -(7.29), \\ & \text{Waterborne AGV dynamics } (3.1) - -(3.5), \\ & \text{Disturbance dynamics } (3.4), \\ & \text{Control constraints } (3.3), (3.8), \end{aligned} \tag{7.30}$$

where the total cost J depends on scheduling cost J_s and control cost J_c. The optimal scheduling and control decisions $\mathbf{u}_\mathrm{s}^*$, $\mathbf{u}_\mathrm{c}^*$ are simultaneously made by solving problem **P**. Apparently, the decision frequency of problem **P** should be the same as the control problem which has faster decision frequencies. However, the already complex scheduling problem (7.1)–(7.29) coupled still with lower level motion and disturbance models will typically not be easily solved to optimality at a high frequency. Even if we decompose problem **P** hierarchically and solve scheduling and control problems sequentially every time a control decision is implemented and new system states are available, solving the scheduling problem at the control frequency could still preclude it from practical applications. We propose an interaction model and a real-time scheduling problem that enable solving scheduling and control problems both hierarchically and real-time in a closed-loop way.

7.4.1 Modeling interactions and real-time speed assignment

Literature dealing with scheduling or control problems independently can actually be viewed as considering a simplified problem **P** with assumptions. From a scheduling perspective, implicit assumptions are made that schedules are all executed perfectly, i.e., waterborne AGVs depart and arrive at berths following the exact scheduled order and timing irrespective of operational disturbances. This is implicitly achieved by assuming a constant speed and thus constant travel times on all routes if speed is not an optimization variable, e.g., in [96], or variable speeds as functions of discrete events if a combined route-speed optimization problem is solved, e.g., in [7, 25, 92] as well as our scheduling problem (7.1)–(7.29). No

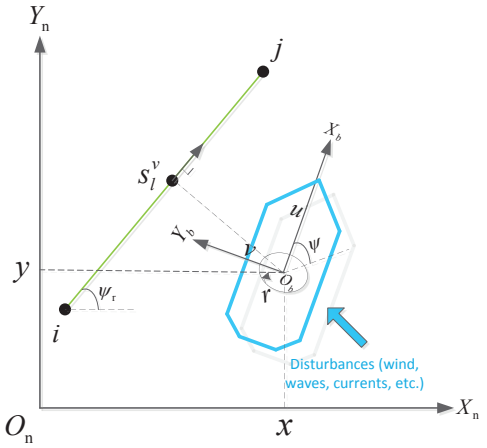


Figure 7.2: Waterborne AGV v and pose projection in route (i, j).

operational disturbances and physical system limitations can further be incorporated. Therefore, waterborne AGV motions are simply modeled as a first order integrator with constant speeds on one arc in a combined routing-speed optimization problem, i.e.,

$$s_v(k+1) = s_v(k) + u_{i(i+1)v} T_s, \forall i \in \mathcal{N}_v(k), v \in \mathcal{V}(k), kT_s \in [\max(a_i(k), t_{i,\min}), A_{i+1}(k)], \tag{7.31}$$

where s_v is the distance waterborne AGV v has traveled on route i.

We take advantage of the implicit motion model in the scheduling problem and propose an interaction model based on a two-level parameterization of reference paths similar as in Chapter 4. The lower level is embedded in the online control problem which takes care of the waterborne AGV dynamics, system limitations, operational disturbances, and collision avoidance with other traffic, and modeled as:

$$s_v^{\mathrm{l}}(k+1) = s_v^{\mathrm{l}}(k) + u_v^{\mathrm{l}}(k) T_s, \forall v \in \mathcal{V}(k), \tag{7.32}$$

where $s_v^{\mathrm{l}}(k), u_v^{\mathrm{l}}(k)$ are a lower level path parameter and its speed, respectively. Both $s_v^{\mathrm{l}}(k)$ and $u_v^{\mathrm{l}}(k)$ are decision variables in online control optimization problems, as to be introduced further in Section 7.4.2. The path parameter determines the reference orthogonal projections of waterborne AGV v onto its current route $(i, i+1), i \in \mathcal{N}_v$ by

$$\begin{aligned} x_p(k) &= s_v^{\mathrm{l}}(k) \sin(\psi_i) + x_i, \\ y_p(k) &= s_v^{\mathrm{l}}(k) \cos(\psi_i) + y_i, \end{aligned} \tag{7.33}$$

as shown in Figure 7.2, with $(x_p(k), y_p(k))$, (x_i, y_i) the inertial frame coordinates of the reference projection and node i, respectively.

The upper level of the two-level parameterization scheme is a partial scheduling problem of problem (7.1)–(7.29) and updates schedules based on the lower level states which in turn reflect waterborne AGV operational details including real-time control performances, delays caused by environmental disturbances or collision avoidance, etc. In this way, the two-level interaction model connects scheduling and control problems but make them decomposable

from each other while still allowing both to be solved online using real-time feedback. The upper level problem is formulated as a real-time speed assignment problem as follows.

At each time step k, we collect feedback information for the set of waterborne AGVs $\mathcal{V}(k)$ that are scheduled with tasks but have not arrived at the scheduled last node each with load $y_v^0(k), v \in \mathcal{V}(k)$ on board. For each waterborne AGV $v \in \mathcal{V}(k)$, the list of yet to visit nodes are $\mathcal{N}_v(k)$, the corresponding list for time windows $\left\{[t^v_{\min,i}, t^v_{\max,i}]\right\}, \forall i \in \mathcal{N}_v(k)$ and the list of service times $\{s_i^v\}, \forall i \in \mathcal{N}_v(k)$. Besides time window constraints, different waterborne AGVs still need to coordinate their service time slots at a particular berth by guaranteeing a safe time interval among them. Variables related to the real-time speed assignment problem are:

- Binary variables: $I_{ij}(k)$ for $i, j \in \mathcal{C}_b, i \in \mathcal{N}_p, j \in \mathcal{N}_q, p \neq q$ equal to 1 if node i is visited by waterborne AGV p before node j by waterborne AGV q and 0 otherwise;
- Continuous variables: $a_i^v(k)$ for $i \in \mathcal{N}_v, v \in \mathcal{V}(k)$ specifies the arrival time of waterborne AGV v at node i;
- Continuous variables: $w_i^v(k)$ for $i \in \mathcal{N}_v, v \in \mathcal{V}(k)$ is the waiting time of waterborne AGV v at node i;
- Continuous variables: $d_i^v(k)$ for $i \in \mathcal{N}_v, v \in \mathcal{V}(k)$ is the delay time of waterborne AGV v at node i;
- Continuous variables: $u^v_{i(i+1)}(k)$ for $i \in \mathcal{N}_v, v \in \mathcal{V}(k)$ is the speed waterborne AGV v travels at on leg $i \to i+1$.

The overall goal is to compute schedules that still minimize the overall cost of fulfilling all remaining requests while satisfying time window and coordinated berthing constraints. The mixed integer programming problem is formulated as:

$$\min\ c_3 \sum_{(i,j)\in\mathcal{A}} u_{ij}^2(k) d_{ij}(k) + c_4 \left\| A_{\mathcal{V}_e}(k) - A_{\mathcal{V}_o}(k) \right\|_1 + c_5 \|\mathbf{w}(k)\|_1 + c_6 \|\mathbf{d}(k)\|_1 \tag{7.34}$$

subject to

$$\max(a_i(k), t^v_{i,\min}) + s_i^v + d_{i(i+1)}(k)/u_{i(i+1)}(k) = A_{i+1}, \quad \forall i \in \mathcal{N}_v(k), v \in \mathcal{V}(k), \tag{7.35}$$

$$t^v_{i,\min} - w_i^v(k) \leqslant a_i(k) \leqslant t^v_{i,\max} - s_i^v + d_i^v(k), \quad \forall i \in \mathcal{N}_v(k), v \in \mathcal{V}(k), \tag{7.36}$$

$$0 \leqslant w_i^v(k) \leqslant w_{\max}, \quad \forall i \in \mathcal{N}_v(k), v \in \mathcal{V}(k), \tag{7.37}$$

$$0 \leqslant d_i^v(k) \leqslant d_{\max}, \quad \forall i \in \mathcal{N}_v(k), v \in \mathcal{V}(k), \tag{7.38}$$

$$I_{ij}(k) + I_{ji}(k) = 1, \quad \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, \; i \in \mathcal{N}_p, j \in \mathcal{N}_q, p \neq q, \tag{7.39}$$

$$I_{ij}(k) = 1 \Rightarrow \max(a_i, t^v_{i,\min})(k) + s_i^v(k) + T \leqslant A_j, \quad \forall i, j \in \mathcal{C}_b, b \in \mathcal{B}, \; i \in \mathcal{N}_p, j \in \mathcal{N}_q, p \neq q, \tag{7.40}$$

$$\underline{u} \leqslant u^v_{i(i+1)}(k) \leqslant \bar{u}, \quad \forall i \in \mathcal{N}_v, v \in \mathcal{V}(k), \tag{7.41}$$

$$I_{ij}(k) \in \{0,1\} \quad \forall (i,j) \in \mathcal{A}, v \in \mathcal{V}, \tag{7.42}$$

where objective (7.34) contains four terms that are the same as the last four terms in (7.1), minimizing energy consumption due to variable speed, total sojourn times, waiting times, and delay times, respectively. Constraints (7.35) indicates the time consistency between two successive nodes in the node list of each waterborne AGV $v \in \mathcal{V}(k)$. Soft time windows are imposed by constraints (7.36) with constraints (7.37) and (7.38) specifying the

maximum waiting and delay times, respectively. Constraints (7.39) and (7.40) together formulate the coordinated berthing times between different waterborne AGV visiting a same berth. Variable speeds are bounded by constraint (7.41) and constraint (7.42) define the only binary variable in this problem. By solving this problem, we obtain updated schedules $\left\{\mathcal{N}_v(k), \mathcal{A}_v(k), \mathcal{Q}_v(k), \mathcal{U}_v(k)\right\}$ for waterborne AGVs $v \in \mathcal{V}(k)$ as well as parameterized reference paths at the upper level, defined as:

$$s_v^{\mathrm{u}}(k+1) = s_v^{\mathrm{u}}(k) + u_{i(i+1)}(k)T_s, \forall v \in \mathcal{V}(k), i \in \mathcal{N}_v(k), \tag{7.43}$$

where $s_v^{\mathrm{u}}(k), u_{i(i+1)}(k)$ are references for $s_v^{\mathrm{l}}(k), u_v^{\mathrm{l}}(k)$, respectively.

7.4.2 Closing the real-time loop

With the interaction model, we are now ready to decompose problem **P** hierarchically into a scheduling problem **Ps**

$$\begin{aligned} \mathbf{u}_{\mathrm{s}}^* &:= \arg\min J_{\mathrm{s}} \\ \text{Subject to} \quad & \text{Scheduling constraints } (7.2) - -(7.29) \text{ or } (7.35) - -(7.42), \end{aligned} \tag{7.44}$$

and a control problem **Pc**

$$\begin{aligned} \mathbf{u}_{\mathrm{c}}^* &:= \arg\min J_{\mathrm{c}} \\ \text{Subject to} \quad & \text{Waterborne AGV dynamics } (3.1) - -(3.5), \\ & \text{Disturbance dynamics } (3.4), \\ & \text{Control constraints } (3.3), (3.8). \end{aligned} \tag{7.45}$$

In particular, at each control time step k, we first solve scheduling problem **Ps** based on updated feedback information: the set of waterborne AGVs $\mathcal{V}(k)$ that are scheduled with tasks but have not arrived at the scheduled last node, each with load $y_v^0(k), v \in \mathcal{V}(k)$ on board, projected waterborne AGV positions $\left(x_p^v(k), y_p^v(k)\right) \forall v \in \mathcal{V}(k)$ that are determined by measured waterborne AGV positions $x_v k, y_v(k) \forall v \in \mathcal{V}(k)$. The projected positions are utilized to *1*) update the Euclidean distances $d_{i(i+1)}(k) \forall i \in \mathcal{N}_v(k), v \in \mathcal{V}(k)$; and *2*) initialize both levels in the interaction model by $s_v^{\mathrm{l}}(k) = s_v^{\mathrm{u}}(k) = \left\| \begin{array}{c} x_p^v(k) - x_i \\ y_p^v(k) - y_i \end{array} \right\|_2$. Note that only at the beginning of the planning horizon, problem (7.1) –(7.29) needs to be solved as the scheduling problem **Ps**. Real-time scheduling is achieved by solving problem (7.34) – (7.42) with a smaller set of integer variables and is efficient to solve to optimality.

At each time step k, the control problem **Pc** is solved after receiving references from scheduling problem **Ps**. The overall control goals are to *1*) execute schedules to fulfill ITT requests in an economical way; *2*) maintain safety by satisfying system physical limitations and avoiding collisions with other traffic in the presence of disturbances; and *3*) maneuver in a distributed way. We present the formulation of J_{c} here which involves the lower level of the interaction model in the closed-loop scheduling and control:

$$J_c(k) = \sum_{v \in \mathcal{V}(k)} \left(c_7 \|\boldsymbol{\eta}(k) - \boldsymbol{\eta}_{\mathrm{r}}(k)\|_2^2 + c_8 \left\| s_v^{\mathrm{l}}(k) - s_v^{\mathrm{u}}(k) \right\|_2^2 + c_9/2 \|\mathbf{v}(k)\|_{\boldsymbol{M}(k)}^2 \right), \tag{7.46}$$

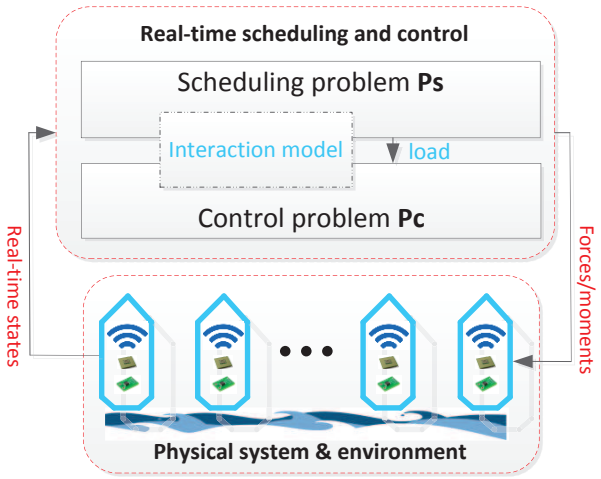


Figure 7.3: Closed-loop scheduling and control of energy-efficient waterborne AGVs.

where η_r is reference pose determined by (7.32) and (7.33) as $\eta_r = \begin{bmatrix} x_p & y_p & \psi_i \end{bmatrix}^T$. Control performance is also affected by scheduling results reflected in the mass matrix as:

$$\boldsymbol{M}(k) = \begin{bmatrix} m + m_c y_v^0(k) & 0 & 0 \\ 0 & m + m_c y_v^0(k) & m + m_c y_v^0(k) x_g \\ 0 & m + m_c y_v^0(k) x_g & I_z \end{bmatrix} + \boldsymbol{M}_A, \tag{7.47}$$

The first goal of executing schedules in an energy efficient way is then achieved by minimizing $J_c(k)$ as in Chapter 5. The closed-loop schedule and control of waterborne AGVs is shown in Figure 7.3.

7.5 Experiments and discussion

Simulations are run to illustrate the effectiveness of the proposed closed-loop scheduling and control of waterborne AGVs for ITT. Note that in practice, the number of ITT requests and the fleet of waterborne AGVs could be large. However, as simplified in Section 7.2, i.e., requests in different planning horizons are decoupled so that small sets of requests can be considered independently. Scheduling problems for each set can be solved repetitively using the proposed approach. The idea is to demonstrate how the algorithm works as well as its effectiveness. Therefore, an ITT scenario as shown in Figure 7.1 with six berths and a fleet of three waterborne AGVs ($V_i, i = 1,2,3$) is considered in simulations in this chapter. Assume that there are seven ITT requests arising within the scheduling horizon 0 s – 2100 s as detailed in Table 7.1 of which the available request information structure has been designed according to [118]. Positions of the six berths are determined in latitude/longitude and subsequently converted to coordinates in inertial frame with Berth 1 as the origin. The fleet of available waterborne AGVs for this set of ITT requests are initially all positioned at Berth 1.

Speed limits of waterborne AGVs are $u_{\min} = 2.57$ m/s, $u_{\max} = 6.68$ m/s which are corresponding to 5 knots and 13 knots, respectively. Each waterborne AGV can carry a maximum

Table 7.1: ITT requests to be carried out.

Request ID	Origin	Destination	Release time (s)	Due time (s)	Volume (TEU)
1	5	2	125	1865	2
2	1	3	690	1155	2
3	1	4	700	1485	1
4	6	2	725	1535	2
5	6	1	1230	1755	2
6	2	3	1345	1750	2
7	1	4	1640	2085	1

of four TEUs, i.e., $Q = 4$, and each TEU of container weighs $m_c = 24000$kg. Each move of a quay crane can load/unload one or two TEUs and requires 120s [129]. Therefore, for all ITT requests in Table 7.1, service times are the same as $t_s = 120$s. The necessary safety time interval between different waterborne AGVs visiting a same berth is set to $T = 60$s based waterborne AGV lengths and sailing speeds. Other parameters concerned with waterborne AGV dynamics are implemented as in [150]. The weight parameters in cost functions (7.1), (7.34) and (7.46) for trade-offs of different performance metrics are set as: $c_1 = 10^4, c_2 = 10^{-2}, c_3 = 10^2, c_4 = 10^3, c_5 = 10^8, c_6 = 10^8, c_7 = 100, c_8 = 100, c_9 = 1$. Algorithms are implemented in MATLAB 2011b [75]. Optimization problems are solved by Cplex [46]. All the simulations are run on a platform with Intel (R) Core (TM) i5-3470 CPU @3.20 GHz.

The closed-loop schedule and control algorithm as shown in Figure 7.3 needs to replace human operators to make "smart" decisions:

- For the fleet of waterborne AGVs, energy efficient schedules as well as actuator inputs to execute these schedules should be autonomously generated satisfying waterborne AGV physical limitations, e.g., maximum capacity, rudder force range etc. and guaranteeing safety;
- For the list of ITT requests, certain amount of containers should be transported from specified origins to destinations after the release time while before the due time; and
- Each berth can accommodate at most one waterborne AGV, and service time slots of different waterborne AGVs should keep a buffer time interval.

Simulation results from these three perspectives are presented next to demonstrate the effectiveness of the proposed closed-loop scheduling and control algorithm.

7.5.1 From the waterborne AGV perspective

The set of seven ITT requests as shown in Table 7.1 calls for all three waterborne AGVs of which initial schedules by solving problem (7.1) – (7.28) are shown in Figure 7.4 as green-circle line, magenta-hexagram dotted line, and green-square dashed line, respectively. All three waterborne AGVs start from Berth 1 but at different times. The small rectangles are one TEU containers and the numbers attached identify IDs of requests that the containers belong to. We display the set and mix of containers on board waterborne AGVs departing

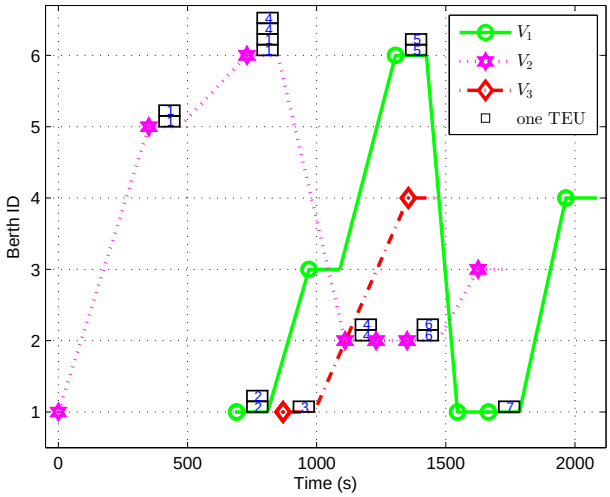


Figure 7.4: Initial energy efficient schedules.

berths. Each schedule contains information on the sequence of berths to visit, the corresponding arrival and departure times as well as the load/unload operations at each berth. Take the schedule of waterborne AGV V_2 as an example, we place a hexagram upon waterborne AGV V_2's arrival at a berth. There are three hexagrams at Berth 2 between 1000s and 1500s because waterborne AGV V_2 performs three load/unload operations at Berth 2. From the set and mix of containers on board when departing, we can derive that waterborne AGV V_2 first unloads the two containers from request 1 taking 120s, then unloads the two containers from request 4 taking another 120s and finally loads the two containers from request 6 before departing from Berth 2 to Berth 3 which is its final destination. Note that the time for solving the problem (7.1) – (7.28) depends on the numbers of waterborne AGVs and requests, and tightness of imposed time windows. It takes 127 s in our case, and could be even more time consuming for larger problems.

Travel speeds along all route segments are also explicitly optimized. In fact, with berth IDs, arrival and departure times known, travel speeds can easily be derived from 7.4. The travel speed profile for waterborne AGV V_2 along its route is shown as Figure 7.5. As can also be observed in Figure 7.4, all three waterborne AGVs carry no more than four TEU containers. Figure 7.6 further shows the total number of containers on board throughout the simulation which are all within the maximum capacity of four TEU containers.

Waterborne AGVs receiving schedules as shown in Figure 7.4 are then controlled with a first goal of guaranteeing operational safety and a secondary goal of executing those schedules. Since complex system dynamics, physical limitations, disturbances and collision avoidance between moving waterborne AGVs are not considered in the scheduling problem, real-time waterborne AGVs do not necessarily behave safely and as scheduled: following the scheduled route at specified speed and arriving at scheduled berths at specified times. Figure 7.7 shows the evolutions of velocities of three scheduled waterborne AGVs, respectively. Velocities in three degrees of freedom: surge, sway and yaw are all within safe

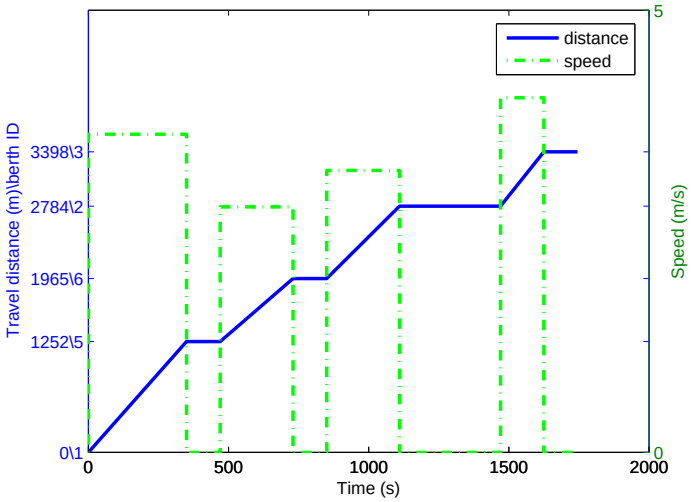


Figure 7.5: Travel speed profile of waterborne AGV V_2.

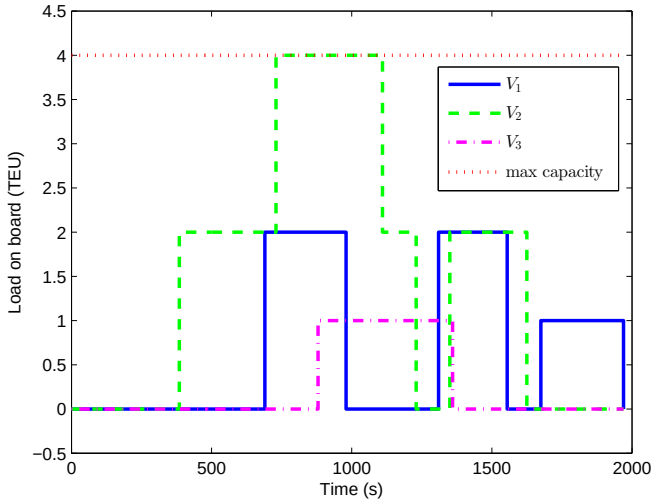


Figure 7.6: Containers on board of the waterborne AGVs.

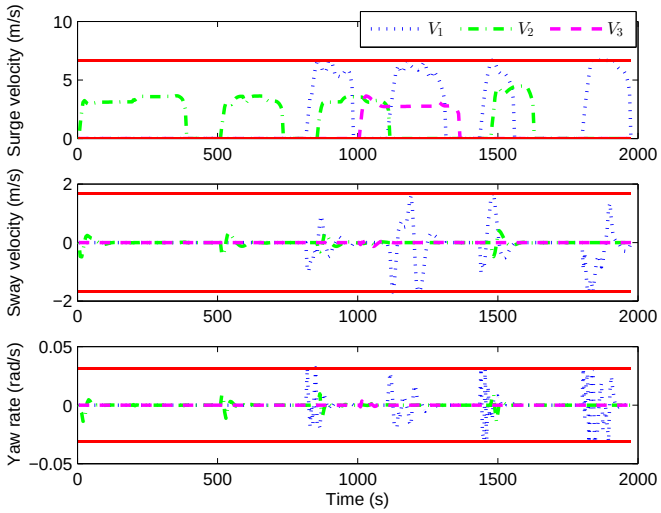


Figure 7.7: Velocities in surge, sway and yaw of three waterborne AGVs.

maneuvering ranges as the red lines show. Surge velocity is a function of time and sees fluctuations which is different with piecewise constant speeds determined in scheduling problems which are functions of events as shown in Figure 7.5. This is due to the necessary accelerations and decelerations when operating in real environment. Sway velocity and yaw rate are not considered in scheduling problems at all. Likewise, control inputs in surge, sway and yaw interact with complex system dynamics and environment to achieve control goals and are all within the safety limitations as shown in Figure 7.8.

Waterborne AGVs are treated equally as agents that are controlled in a distributed way and make control decisions parallelly using a fast distributed control method proposed in [148]. During the execution of assigned requests, waterborne AGVs might encounter conflicts with each other. The distributed control algorithm guarantees ITT request fulfillment locally for each waterborne AGV while achieves an overall minimal cost and safety for all waterborne AGVs. The route following performance with small tracking errors of waterborne AGV V_3 is demonstrated as in Figure 7.9(a). Fluctuations are seen at the beginning and around 1200s due to the start up and encountering with waterborne AGVs V_2 and V_1. However, a safety distance away from them is ensured by the control level when waterborne AGVs are in close range. Figure 7.9(b) shows the distances between waterborne AGV V_2 and the other two waterborne AGVs which are all above the minimum safety distances.

7.5.2 From the ITT request perspective

Assignment of requests in Table 7.1 to three waterborne AGVs can be derived from the request IDs attached with containers in Figure 7.4. This assignment is produced being aware of constraints as waterborne AGV capacities and request time windows as well as travel cost specified in (7.1). For each request, Figure 7.10 shows the time window (red bar) specified by the release and due times of the request, the initial scheduled duration time (yellow bar)

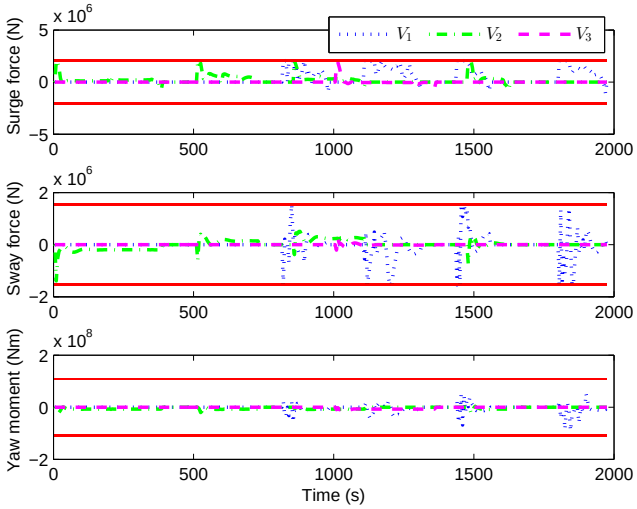


Figure 7.8: Control input evolution in surge, sway and yaw of three waterborne AGVs.

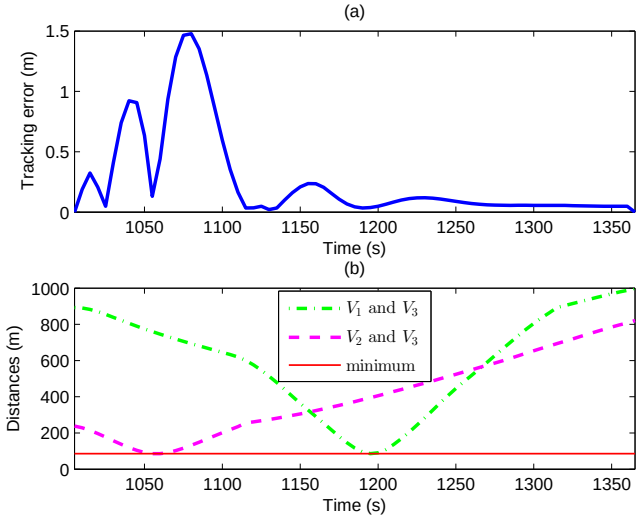


Figure 7.9: Tracking performance and collision avoidance.

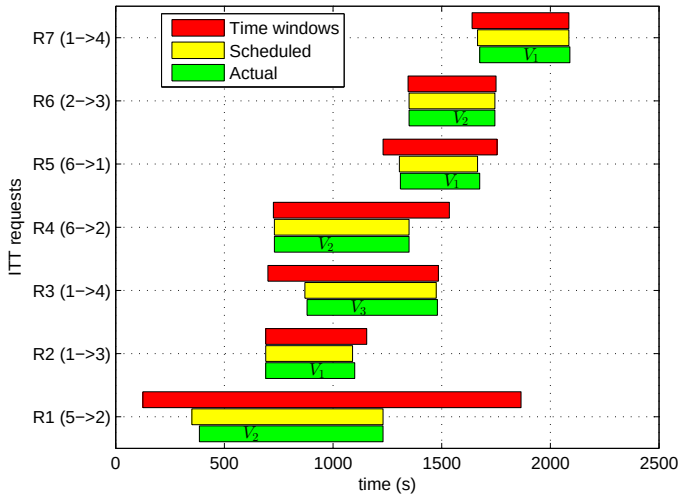


Figure 7.10: Satisfaction of request time windows in the closed-loop.

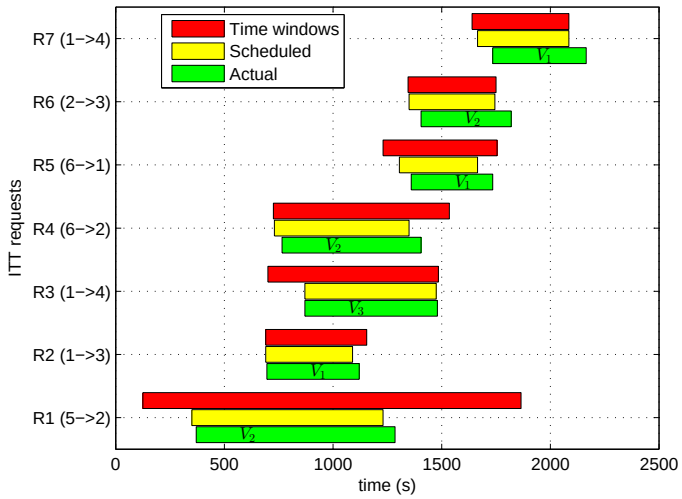


Figure 7.11: Requests executed in the open-loop.

solving problem (7.1) – (7.29) and the actual duration time (green bar) specified by the waterborne AGV's arrival at the origin berth and departure from the destination berth. All the scheduled and actual duration times are within required time windows, i.e., time windows of all requests are satisfied by the scheduling problem and the control problem succeeds in operating waterborne AGVs with timing window awareness except for request 3 with some delays. Some green bars, e.g., request 1, however, are not within the corresponding yellow bars. This is due to the real-time update of schedules by solving closed-loop scheduling problem (7.34) – (7.42). The updated scheduled duration times are not necessarily the same with the initial schedules, but still are guaranteed to satisfy time window constraints as (7.36). The satisfaction of time windows by actual duration times proves this.

For comparison, Figure 7.11 shows the request completion times of schedules executed in open-loop, i.e., the initial schedule by solving (7.1) – (7.28) is not updated. Some requests with relatively tight time windows, e.g, request 6 and 7, see delays. This is because lower level system details that may cause inaccurate execution of schedules are neither considered in scheduling problem (7.1) – (7.28) nor reflected in updated schedules in solving real-time problem (7.34) – (7.42). The inaccuracies accumulate along routes and lead to delays. Waterborne AGV 1, for example, serves request 2, 5, and 7 in sequence (see Figure 7.4). Delays in finishing request 2 lead to later start of request 5 compared to the scheduled start time, and give rise to the violation of the tight time window of request 7 in the end. Since "non-performance", which happens when there are containers that are delivered with delays, is the most important criterion for ITT, we define the "non-performance" rate as the percentage of delayed number (in TEUs) of ITT containers with respect to the total number of ITT containers. Therefore, for the seven ITT requests with 12 TEUs in Table 7.1, the closed-loop approach has a "non-performance" rate of 0% while the open-loop approach is 41.67%. This illustrates the advantage of closed-loop over open-loop.

7.5.3 From the berth perspective

Each of the six waterborne AGV berths as shown in Figure 7.1 is designed to handle maximum one waterborne AGV one time and there should be a time interval between different waterborne AGV's visits to a same berth for berth practice and guaranteeing safety considering dimensional waterborne AGVs. In other words, the departure time of one waterborne AGV should be at least earlier the specified time interval $T = 60$s than the arrival time of the next waterborne AGV as constrained by (7.13) - (7.14). From the berth perspective, Figure 7.12, 7.13, and 7.14 illustrate berth occupations over time as initially scheduled, real-time scheduled at $t = 750$s, and the actually executed by waterborne AGVs, respectively.

For all berths, there are intervals between bars in different colors along one horizontal line representing different waterborne AGVs visiting a same berth in all three figures. Particularly in Figure 7.14 the for actual berthing time slots, Berths 1, 3, 4, and 6 see visits from different waterborne AGVs. The minimum actual time interval is $60\text{s} \geqslant T$ (minimum safety time interval) which happens between waterborne AGVs 1 and 3 at Berth 1. For one waterborne AGV performing more than one load/unload operation at one berth, however, there is no time interval. This is shown by the bars in same colors linked together, e.g., the three magenta bars at Berth 2 in Figure 7.12. Actually, waterborne AGV V_2 performs three load/unload operations at Berth 2 if we recall Figure 7.4 as analyzed in Section 7.5.1. Note that the vertical lines at Berth 1 in Figure 7.12 and Figure 7.14 indicates that water-

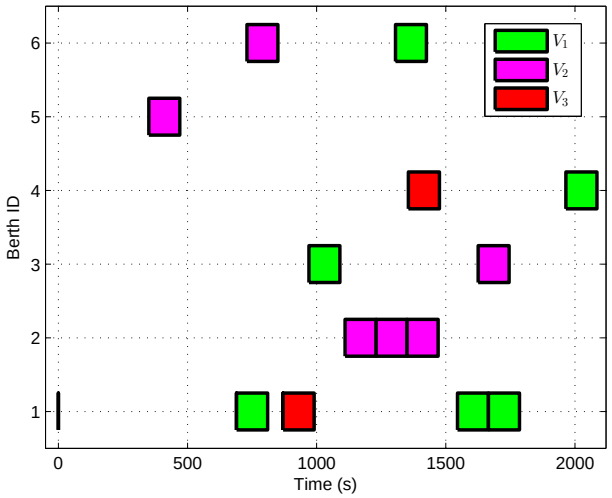


Figure 7.12: Berthing time slots from the initial schedule.

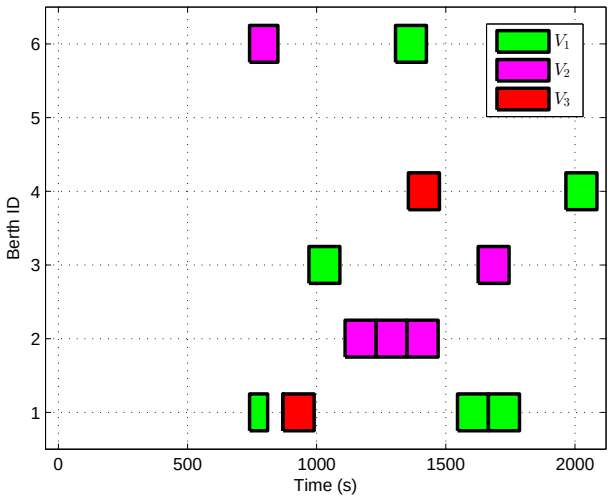


Figure 7.13: Berthing time slots from the schedule at $t = 750s$.

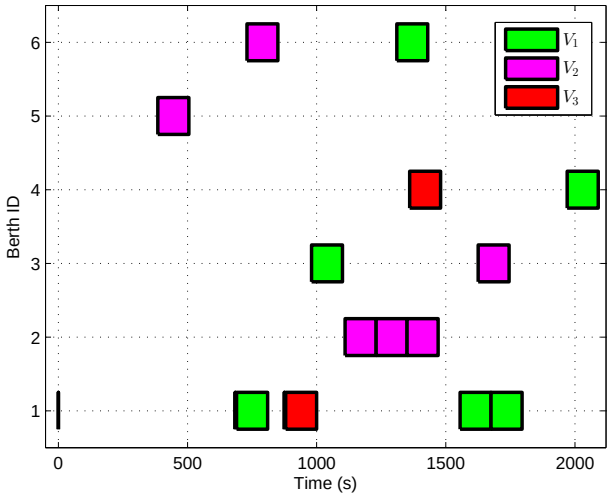


Figure 7.14: Actual berthing time slots.

borne AGV V_2 travels from Berth 1 to Berth 5 directly without any load/unload operations at Berth 1. Lines instead of bars, i.e., no load/unload operations, may only arise at waterborne AGVs' initial positions (Berth 1 in our scenario) since any other berths waterborne AGVs to visit are involved an ITT request with certain amount of containers to load/unload and thus requires certain service times. Generally, all time slots (bars) should be not shorter than $t_s = 120$s. The shorter time slots in Figure 7.13, e.g., as shown by the relatively shorter green bar at Berth 1 and the first magenta bar at Berth 6, are because certain amount of service has been conducted at $t = 750$s. A bar longer than $t_s = 120$s simply means waterborne AGV waits for some time before the load service can start (release time).

7.6 Conclusions

In this chapter, we have proposed a real-time closed-loop scheduling and control scheme for waterborne AGVs applied to ITT. The contributions are twofold. Firstly, we propose a new pick-up and delivery scheduling model for ITT using waterborne AGVs by considering safety time intervals between their service time slots at one berth. For all the berths, safety time intervals are guaranteed for different waterborne AGVs. Secondly, we propose a partial scheduling problem that is efficient to solve. By integrating this partial scheduling problem with the control problem of multiple waterborne AGVs in Chapter 5, we realize real-time closed-loop scheduling and control of an autonomous ITT system. In our simulation experiments based on a potential ITT scenario in the port of Rotterdam, time windows of all ITT requests are satisfied by the closed-loop approach with 0% "non-performance" rate compared to 41.67% from the open-loop approach. The proposed algorithm provides an effective way realizing autonomous ITT systems using waterborne AGVs.

Chapter 8

Conclusions and future research

In this thesis, a new type of container transporter, waterborne AGVs is proposed, and advanced control and scheduling strategies for efficiently operating such waterborne AGVs are discussed. Waterborne AGVs are coordinated for carrying out Inter Terminal Transport (ITT) tasks in port areas. This final chapter presents the conclusions and the main contributions of this thesis in Section 8.1, and gives recommendations for future research on waterborne AGVs in Section 8.2.

8.1 Conclusions and contributions

In this section, we first conclude the thesis by addressing the main research goal and answering the five Key Research Questions raised in Chapter 1 using the proposed strategies from the previous chapters. Contributions of this PhD research with respect to coordination methodologies and logistics applications are then explicitly discussed.

8.1.1 Answering the research questions

As presented in Chapter 1, the main research goal of this thesis is to *develop advanced control and scheduling strategies for coordinated waterborne AGVs applied to ITT*. In Chapters 2 – 7, relevant literature has been reviewed and novel systematic methods have been proposed to achieve the aforementioned goal. Simulation results of ITT case studies using waterborne AGVs have demonstrated the effectiveness of the proposed approaches. More specifically, the five Key Research Questions for addressing the main research goal are answered as follows:

1. *Which technique is suitable for the control of waterborne AGVs?*

   We propose that the control of waterborne AGVs carrying out ITT tasks, as in Chapters 4 – 7, uses model predictive control (MPC). MPC can look into the future and take actions at an earlier stage by predicting system trajectories over a future horizon. Advantages of MPC include handling system constraints explicitly and optimizing multiple objectives quantitatively. Different ITT scenarios with multiple possibly conflicting control goals are investigated. In Chapter 4, one single waterborne AGV with an assigned ITT

task is controlled to track reference paths smoothly and arrive at the specified terminal at the specified time in an energy-efficient way. Overshoots during the switching of reference line segments are avoided. Various constraints such as physical limitations and obstacle avoidance are also satisfied. Chapter 5 considers multiple waterborne AGVs, each of them assigned with an ITT task. In the proposed cooperative distributed MPC (DMPC) framework, waterborne AGVs are able to solve local problems in parallel while possible collisions between waterborne AGVs are avoided and the overall cost is minimized. A cost-effective robust DMPC (RDMPC) algorithm is proposed in Chapter 6 for waterborne AGVs maneuvering in uncertain environments. Cost-effective robustness optimizes the trade-off between system performance and robustness level considering system and uncertainty characteristics. In Chapter 7, the DMPC approach of Chapter 5 is utilized in a closed-loop scheduling and control scheme for multiple waterborne AGVs. All the control problems consider detailed waterborne AGVs dynamics as modeled in Chapter 3. In MPC, successively linearized models are used for prediction to reduce the possible computational complexity. Various simulation experiments have been carried out using the proposed MPC-based algorithms in different ITT case studies. Simulation results show that control goals specified are all successfully achieved, which validates the suitability of MPC in controlling waterborne AGVs for ITT.

2. *What performance criteria should be considered in optimizing the process of one waterborne AGV carrying out one ITT task and how can the optimal performance be achieved?*

   For the most fundamental scenario of one waterborne AGV carrying out one ITT task, smooth reference path tracking, arriving at the destination as punctually as possible, and energy-efficiency should be considered as the performance criteria. Geometric shortest reference paths are given as connected straight-line segments from which derivations of waterborne AGV trajectories are minimized. Waterborne AGVs arrive at the destinations punctually or with a minimal delay to lower the "non-performance" rate of ITT. Energy consumption related to the waterborne AGV speed has also been minimized. These performance criteria are in alignment with the practice of ITT and the requirements in port areas. A predictive path following with arrival time awareness (PPF-ATA) controller has been proposed to satisfy those criteria in Chapter 4. Waterborne AGV kinematics are remodeled in a connected path coordinate in which cross-track and along-track errors are defined. Cross-track errors are minimized to achieve the reference path convergence, and along-track errors that facilitate a path switching logic for avoiding overshoots are minimized to achieve arrival time awareness. Smooth tracking and timing have been guaranteed by a two-level double integrator scheme. Simulations have shown that waterborne AGVs achieve the specified performance criteria with the proposed PPF-ATA controller in two ITT scenarios.

3. *How can multiple waterborne AGVs be coordinated for multiple ITT tasks with waterborne AGVs making decisions locally while minimizing the overall cost in a cooperative and distributed way?*

   In Chapter 5, cooperative distributed PPF-ATA controllers have been proposed for multiple waterborne AGVs carrying out multiple ITT tasks. In such a scenario, couplings between waterborne AGVs in a neighborhood arise as collision avoidance constraints as modeled in Chapter 3. An iterative decomposition-coordination technique, the Alternat-

ing Direction Method of Multipliers (ADMM), has been proposed to achieve cooperative distributed control of waterborne AGVs in the framework of DMPC. Furthermore, the possible poor convergence rates of conventional ADMM are improved by a proposed fast ADMM approach by iteratively approximating global information in local problems. In our ITT case study in the port of Rotterdam, comparing ADMM and fast ADMM, ADMM requires 108 iterations with a total solver time of 4.65s before convergence while fast ADMM requires only six iterations with 0.27s. Therefore, fast ADMM offers a more practical cooperative distributed approach considering the short sampling times of waterborne AGVs. For both ADMM and fast ADMM distributed control approaches, all waterborne AGVs can fulfill assigned ITT tasks successfully. Cooperative and safe behaviors have been observed in conflicting areas. Therefore, the proposed cooperative distributed controllers are effective in coordinating multiple waterborne AGVs for multiple ITT tasks.

4. *How can environmental disturbances due to wind, waves, and current be systematically handled by cooperative and distributed waterborne AGVs?*

   The influences of environmental disturbances have been considered in two scenarios: one with perfectly known disturbances, and the other with not perfectly known but roughly predicted disturbances, as modeled in Chapter 3. If environmental disturbances are perfectly known, the cooperative DPPF-ATA controllers proposed in Chapter 5 are readily applicable. For roughly predicted stochastic disturbances, a cost-effective RDMPC approach for multiple waterborne AGVs has been proposed in Chapter 6. The approach is cost-effective in the sense that the overall system robustness level and the associated price of robustness are explicitly optimized considering system and uncertainty characteristics. Since probabilistic distributions of uncertainties are parameterized by introducing binary variables, the convexity assumptions for the cooperative distributed control in Chapter 5 do not necessarily hold. An efficient integrated branch & bound (B&B) and ADMM algorithm has been proposed to solve the cost-effective RDMPC problem. The algorithm exploits the special ordered probability sets conducting smart search in B&B and integrates branching criteria with intermediate ADMM results for early termination. With the proposed efficient searching strategy, at most two branching operations at each time step are required throughout the simulation. Simulation results show that, similar with the deterministic cases in Chapter 5, all waterborne AGVs fulfill assigned ITT tasks successfully and safely in uncertain scenarios as well. Cooperative distributed decision making has also been achieved following ADMM iterations. The trade-off between robustness level and the price of being robust has been optimized systematically. In our simulations, all robustness levels are higher than 95% to ensure high safety level. Therefore, the cooperative DPPF-ATA control strategy proposed in Chapter 5 and the cost-effective RDMPC approach proposed in Chapter 6 have provided effective ways for handling environmental disturbances for multiple waterborne AGVs.

5. *In what way can the scheduling and control loop for waterborne AGVs be closed in order to obtain an energy-efficient autonomous ITT system?*

   An energy-efficient autonomous ITT system using waterborne AGVs has been realized in Chapter 7 with a closed-loop scheduling and control approach. Both the scheduling and control problems have been solved in real-time with updated system states. Deci-

sions are made hierarchically to guarantee tractability. However, factors such as unconsidered physical system limits, disturbances, and collision avoidance that are difficult, if not impossible, to be integrated in a scheduling problem can be reflected timely in the online updated schedules. A new coordinated berthing time schedule model considering necessary time intervals between different waterborne AGVs visiting a particular berth has been proposed. For all the berths, safety time intervals are guaranteed for different waterborne AGVs. Furthermore, a partial scheduling problem that is efficient to solve, and an interaction model that integrates the scheduling and control problems have been proposed. Cooperative distributed waterborne AGVs are controlled using the fast ADMM algorithm proposed in Chapter 5. Given a set of ITT requests and a fleet of waterborne AGVs, the proposed closed-loop schedule and control framework works in an autonomous way. In our simulation experiments based on a potential ITT scenario in the port of Rotterdam, time windows of all ITT requests are satisfied by the closed-loop approach with 0% "non-performance" rate compared to 41.67% from the open-loop approach. The proposed autonomous ITT system using waterborne AGVs is demonstrated to be effective and contributes to smarter port logistics.

8.1.2 Contributions

Having answered the above Key Research Questions, the research presented in this PhD thesis contributes to the state-of-the-art in the following aspects:

- A novel type of container transporter, waterborne AGVs, has been proposed for improving the *port level* automation and efficiency, which also stimulates a new research stream in the fields of control and logistics, see, e.g., [148, 150, 151, 153].
- Systematic advanced control and scheduling strategies have been proposed and demonstrated to be effective for coordinating waterborne AGVs with applications in ITT.

Control and scheduling

Particularly, the contributions of this PhD thesis to the state-of-the-art with respect to control and scheduling approaches are as follows:

- Complex networked nonlinear systems in both certain and uncertain cases have been approximated using a successive linearization approach in the framework of MPC, and trade-offs among computational complexity, optimality, and ease of controller design have been achieved in Chapter 3 – 6 (see also [148, 150, 151]);
- The problem of predictive path following with arrival time awareness has been proposed and solved with a new reference path coordinate system and a two-level parameterization scheme in Chapter 4 (see also [150]);
- Parallel computations in cooperative distributed MPC for time-varying networked systems has been achieved via ADMM and a fast ADMM algorithm with faster convergence rate in Chapter 5 (see also [148, 149]);
- The robustness level of uncertain systems has been proposed, and a cost-effective RDMPC scheme has been proposed for time-varying networked systems with cooperative distributed decision making in Chapter 6 (see also [151, 152]);

- A novel scheduling approach considering safe intervals between berthing time slots of different vehicles visiting the same berth has been proposed in Chapter 7 (see also [153]);
- Real-time scheduling and control loop has been closed by a partial scheduling model and an interaction model with feedback reflecting neglected factors from lower levels in Chapter 7 (see also [153]).

Transport and logistics

With respect to transport and logistics, several typical ITT case studies have been investigated using waterborne AGVs coordinated by the proposed strategies. More specifically,

- A singe ITT task has been achieved using one waterborne AGV with smoothly path tracking, arrival time awareness, and energy efficiency in Chapter 4 (see also [150]);
- A strategy for cooperatively accomplishing multiple ITT tasks is proposed based on coordinating multiple waterborne AGVs in a parallel distributed way in Chapter 5 (see also [148, 149]);
- Cost-effective robust performance has been achieved in fulfilling ITT tasks by multiple waterborne AGVs even when environmental disturbances are present in Chapter 6 (see also [152]); and
- An energy efficient autonomous ITT system using waterborne AGVs has been realized by the proposed closed-loop real-time scheduling and control framework in Chapter 7 (see also [153]).

8.2 Future research

As discussed in Chapter 1, a typical transport system consists of various decision problems at different levels. In principle, mutual interactions exist between different problems and an integrated design is desirable. However, an integrated problem size could be intractable. For the proposed new type of transporter, waterborne AGVs, this thesis has provided several effective coordination strategies for the involved scheduling and control problems. Recommendations for future research are given in this section with respect to the coordination and further development of waterborne AGVs. The aim is to stimulate more awareness of waterborne AGVs in the community, inspire further work on this topic, and advance waterborne AGVs in practical port logistics systems.

8.2.1 Directions for coordinating waterborne AGVs

This section briefly discusses several technically open issues relevant with coordination of waterborne AGVs, and gives recommendations for future work on these aspects.

- *Modeling accuracy.*

  This thesis models the waterborne AGV behavior by the dynamics of marine surface vehicles with three DOFs (surge, sway, and yaw) as presented in Chapter 3. This model

considers detailed hydrostatics, hydrodynamics, and kinematics with environmental influences in the inertial and body-fixed coordinate systems, and involves highly nonlinear terms. The main maneuvering characteristics of waterborne AGVs can be captured by this model. However, in practice, the hydrodynamics can be varying in different occasions. Examples include: *a*) the load, i.e., the number or position of containers, on board is altered. In Chapter 7, the rigid body mass is adapted to the number of containers on board of waterborne AGVs. More accurate hydrodynamic influences should be considered. *b*) the hydrodynamic interactions between waterborne AGVs in the proximity are not modeled. For networked waterborne AGV models, collision avoidance couplings with independent waterborne AGV dynamics are considered. However, hydrodynamic interactions could give rise to couplings also in dynamics. Model-based controller design, e.g., MPC, is expected to achieve better control performance using more accurately modeled waterborne AGVs. In particular, for varying and uncertain parameters in waterborne AGV models, more research work could be done using, e.g., online parameter identification, robust, and adaptive methods.

- *Computational efficiency.*

  In this thesis, scheduling and control problems have been handled hierarchically for a tractable solution. Nonetheless, computational efficiency for both the scheduling and control problems of waterborne AGVs could still be improved.

  The scheduling problems proposed in Chapter 7 for ITT using waterborne AGVs are originally mixed integer nonlinear problems, and transformed into mixed integer linear problems which are still NP-hard. The computational complexity increases exponentially with increasing sizes of the waterborne AGV fleet and ITT requests. Therefore, the full scheduling problem is solved offline handling seven ITT requests within the considered horizon. More dynamic scenarios dealing with larger numbers of ITT requests are expecting either more advanced computational hardware, faster exact optimization solvers, or a fast heuristic solution approach accepting certain optimality gaps.

  For the control problems, since MPC requires solving optimization problems online repetitively, computational efficiency is also critical. Efforts that have been made in this thesis to relieve possibly heavy computational burdens include successive linearizations to convexify optimization problems (Chapter 3), distributed control (Chapter 5), a proposed fast ADMM based distributed control approach (Chapter 5), analytical tube bounds (Chapter 6), and the efficient search in an integrated branch & bound and ADMM approach in robust control (Chapter 6). However, for more complicated scenarios, it can still currently not be guaranteed that decisions are updated timely even with the aforementioned designs. For example, when a large number of waterborne AGVs are involved, it could take a long time before an agreement is reached within the group. In general, sharing more information within the coupling group helps in terms of convergence rates for distributed control algorithms. However, privacy issues might occur. For cases in which timely update and feedback are critical, a reliable decision recovery mechanism, e.g., making safe but not necessarily optimal decisions based on the previous solution, could be necessary. Furthermore, comparisons between the proposed controllers and other advanced or practically implemented controllers with respect to system performance and computational efficiency could also be done in the future work.

- *Coordination between waterborne AGVs with different owners and between waterborne AGVs and manned marine vehicles.*

  Waterborne AGVs considered in this thesis are designed to be cooperative to optimize an overall cost and no priorities are assigned. However, in practice, waterborne AGVs could belong to different terminals and operated by different companies. A more complicated scenario involves manned marine vehicles which cannot be controlled to be cooperative. In these cases, non-cooperative coordination approaches should be investigated. One option is to assign priorities to vehicles according to certain rules, e.g., the International Regulations for Preventing Collisions at Sea (COLREGs) [97]. COLREGs are the navigation rules followed by manned ships to prevent collisions. Flexibility and optimality could be the concerns for the rule-based coordination. More work could be done on integrating rule-based coordination into the proposed coordination strategies in this thesis to handle mixed traffic situations.

8.2.2 Additional directions for future research

Real applications of waterborne AGVs to ITT in port areas still call for more technological, methodological, and legal advances. Key challenges regarding the development of waterborne AGVs are as follows:

- *Cost-benefit analysis.*

  Research on a systematic cost-benefit analysis of applying waterborne AGVs to ITT could be carried out. Data in terms of labor, time, and cost savings, etc., could be collected from port authorities for the cost-benefit analysis. Moreover, analytical and simulation models of an ITT system with waterborne AGVs involved could be built to predict and analyze the potential strengths and weaknesses of the system.

- *Design and construction.*

  Waterborne AGV design and construction could be seen as a naval architecture problem and solved by ship building experts. However, knowledge on transport and logistics should be given as the input. For example, size and capacity of waterborne AGVs can be related with the ITT demand scenarios. In the context of green port logistics, clean power systems should be considered in the design and construction of waterborne AGVs.

- *Supporting infrastructures.*

  Parking, maintenance, charging or refueling infrastructures should be designed and built for Waterborne AGVs. Besides, communication infrastructures including waterborne AGV-to-waterborne AGV and waterborne AGV-to-infrastructure are important in realizing the coordination framework for waterborne AGVs.

- *Interactions with other port logistics processes.*

  As discussed in Chapter 2, currently, ITT is mainly realized by multi-trailer systems. The assignment problem of ITT requests to different transport modes should be extended to consider waterborne AGVs. Interactions with terminal operations, such as the scheduling of quay cranes for loading/unloading waterborne AGVs and the scheduling of land-side AGVs for transporting unloaded containers could also be investigated.

- *Macroscopic marine traffic flow analysis.*

  At the macroscopic level, the impacts of introducing waterborne AGVs to the existing maritime traffic could also be identified. On the one hand, since waterborne AGVs take advantage of advanced information and communication technologies and are operated in an optimal way, the introduction of them might improve maritime traffic efficiency. On the other hand, as an extra type of vehicles with possibly lower capacities than manned barges, the introduce of them might increase the maritime traffic flow. Therefore, it is suggested that the current maritime traffic flow analysis incorporates waterborne AGVs in the models and simulations as well.

- *Industrial awareness and acceptance.*

  More work should be done to raise the awareness of waterborne AGVs in relevant industries. Pilot or experimental platforms should be built for tests and demonstrations.

Bibliography

[1] L. C. Andrews. *Special Functions for Engineers and Applied Mathematicians*. Macmillan, London, UK, 1985.

[2] D. Angeli. Economic model predictive control. In J. Baillieul and T. Samad, editors, *Encyclopedia of Systems and Control*, pages 1–9. Springer London, 2014.

[3] APM Terminals. APM terminals maasvlakte II terminal officially opens. http : / / www . apmterminals . com / news / press-releases / 2014 / 12 / apmt-terminals-maasvlakte2-officially-opens, 2015. Accessed: May 2016].

[4] K. J. Aström and R. M. Murray. *Feedback Systems: An Introduction For Scientists and Engineers*. Princeton university press, New Jersey, USA, 2nd edition, 2015.

[5] M. Baldea and I. Harjunkoski. Integrated production scheduling and process control: A systematic review. *Computers & Chemical Engineering*, 71:377–390, 2014.

[6] E. M. L. Beale and J. A. Tomlin. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In *Operational Research*, volume 69, pages 447–454. Tavistock Publishing, London, 1970.

[7] T. Bektaş and G. Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

[8] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[9] E. A. Boroujeni and H. R. Momeni. Adaptive sliding mode control for roll motions of ships. In *Proceedings of International Conference on Control, Automation and Systems*, pages 1622–1625, Seoul, South Korea, 2008.

[10] J. R. Boyd. Patterns of conflict. Technical report, Washington, DC: U.S. Air Force, 1986. Accessed: October 2016.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge university press, Cambridge, UK, 2004.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[13] G. Bruzzone, G. Bruzzone, M. Bibuli, and M. Caccia. Autonomous mine hunting mission for the charlie USV. In *Proceedings of 2011 OCEANS*, pages 1–6, Santander, Spain, 2011.

[14] S. Campbell, W. Naeem, and G. W. Irwin. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36(2):267–283, 2012.

[15] C. Chen, C.-C. Liao, and C. Chiang. Adaptive neuro-wavelet control for the ship trajectory tracking problem. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–6, Brisbane, Australia, 2012.

[16] M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Maritime transportation. In C. Barnhart and G. Laporte, editors, *Handbooks in Operations Research and Management Science: Transportation*, volume 14, pages 189–284. Elsevier, Amsterdam, 2007.

[17] J. Curcio, J. Leonard, and A. Patrikalakis. SCOUT – a low cost autonomous surface platform for research in cooperative autonomy. In *Proceedings of 2005 OCEANS*, pages 725–729, Washington, USA, 2005.

[18] R. Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, Germany, 4th edition, 2010.

[19] K. D. Do and J. Pan. Robust path-following of underactuated ships: Theory and experiments on a model ship. *Ocean Engineering*, 33(10):1354–1372, 2006.

[20] J. Du, J. Park, I. Harjunkoski, and M. Baldea. A time scale-bridging approach for integrating production scheduling and process control. *Computers & Chemical Engineering*, 79:59–69, 2015.

[21] M. B. Duinkerken, R. Dekker, S. T. Kurstjens, J. A. Ottjes, and N. P. Dellaert. Comparing transportation systems for inter-terminal transport at the maasvlakte container terminals. *OR Spektrum*, 28(4):469–493, 2006.

[22] W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[23] Egemin Automation Inc. Automated guided vehicles (AGVs). http://www.egeminusa.com/pages/agvs/agvs_general.html, 2015. Accessed May 2016.

[24] Erasmus Smart Port Rotterdam. SmartPort/Port research center Rotterdam Delft poster session 2014. http://www.erim.eur.nl/centres/smartporterasmus/news / detail / 3585-smartportport-research-center-rotterdam-delft-poster- \ session-20131-posters-were-presented-in/, February, 2015. Accessed: June, 2016.

[25] K. Fagerholt, G. Laporte, and I. Norstad. Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3):523–529, 2010.

[26] F. Farokhi, I. Shames, and K. H. Johansson. Distributed MPC via dual decomposition and alternative direction method of multipliers. In J. M. Maestre and R. R. Negenborn, editors, *Distributed Model Predictive Control Made Easy*, pages 115–131. Springer, 2014.

[27] M. A. Figliozzi. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):616–636, 2012.

[28] T. I. Fossen, M. Breivik, and R. Skjetne. Line-of-sight path following of underactuated marine craft. In *Proceedings of the 6th IFAC on Manoeuvring and Control of Marine Craft*, pages 244–249, Girona, Spain, 2003.

[29] T. Fossen and T. Perez. Kalman filtering for positioning and heading control of ships and offshore rigs. *IEEE Control Systems Magazine*, 29(6):32–46, 2009.

[30] T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons Ltd., West Sussex, U.K., 2011.

[31] H. P. Geering. *Optimal Control with Engineering Applications*. Springer, Berlin, German, 2007.

[32] E. Gerritse. Analysis for inter terminal transportation demand scenarios for the Maasvlakte I and II in 2030. Technical report, Delft University of Technology, Delft, The Netherlands, 2014.

[33] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49(3):829–833, 2013.

[34] P. Gomes, C. Silvestre, A. Pascoal, and R. Cunha. A path-following controller for the DELFIMx autonomous surface craft. In *Proceedings of 7th IFAC Conference on Manoeuvring and Control of Marine Craft*, Lisbon, Portugal, 2006.

[35] Google. Maasvlakte Rotterdam, 51.962398° N and 4.056800° E, **Google Earth**, August 7, 2013. Accessed: 2014-10-17.

[36] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*. Springer-Verlag London, 2011.

[37] Gurobi Optimization. Gurobi optimizer reference manual, 2012. URL http://www.gurobi.com. Accessed: June, 2016.

[38] N. Harl and S. Balakrishnan. Impact time and angle guidance with sliding mode control. *IEEE Transactions on Control Systems Technology*, 20(6):1436–1449, 2012.

[39] B. He, H. Yang, and S. Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and applications*, 106(2):337–356, 2000.

[40] L. Heilig and S. Voß. A cloud-based SOA for enhancing information exchange and decision support in ITT operations. In R. G. Gonzlez-Ramrez, F. Schulte, S. Voß, and J. A. Ceroni Daz, editors, *Computational Logistics*, pages 112–131. Springer, New York, USA, 2014.

[41] L. Heilig and S. Voß. Inter-terminal transportation: an annotated bibliography and research agenda. *Flexible Services and Manufacturing Journal*, pages 1–29, 2016. doi: 10.1007/s10696-016-9237-7.

[42] C. Hetherington, R. Flin, and K. Mearns. Safety in shipping: The human element. *Journal of safety research*, 37(4):401–411, 2006.

[43] S. Hill and M. Cannon. A potential feedback approach to ecosystem-based management: model predictive control of the antarctic krill fishery. *Commission for the Conservation of Antarctic Marine Living Resources Science*, 20:119–137, 2013.

[44] Y. Ho and T. Liao. Zone design and control for vehicle collision prevention and load balancing in a zone control agv system. *Computers & Industrial Engineering*, 56(1): 417–432, 2009.

[45] J. E. Hopcroft and R. E. Tarjan. Efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372378, 1971.

[46] ILOG. IBM ILOG CPLEX Optimizer, 2010. URL http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/. Accessed: June 2016.

[47] D. Jia and B. Krogh. Min-max feedback model predictive control for distributed control with communication. In *Proceedings of the 2002 American Control Conference*, volume 6, pages 4507–4512, Anchorage, USA, 2002.

[48] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the 1995 American Control Conference*, volume 3, pages 1628–1632, Seattle, USA, 1995.

[49] C. M. Kellett and A. R. Teel. Smooth lyapunov functions and robustness of stability for difference inclusions. *Systems & Control Letters*, 52(5):395–405, 2004.

[50] 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, 2008.

[51] B. Kouvaritakis, M. Cannon, and J. Rossiter. Non-linear model based predictive control. *International Journal of Control*, 72(10):919–928, 1999.

[52] B. Kouvaritakis, M. Cannon, S. V. Raković, and Q. Cheng. Explicit use of probabilistic distributions in linear predictive control. *Automatica*, 46(10):1719–1724, 2010.

[53] Y. Kuwata and J. P. How. Cooperative distributed robust trajectory optimization using receding horizon MILP. *IEEE Transactions on Control Systems Technology*, 19(2): 423–431, 2011.

[54] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger. Safe maritime autonomous navigation with COLREGs using velocity obstacles. *IEEE Journal of Ocean Engineering*, 39(1):110–119, 2014.

[55] Laboratory for Autonomous Marine Sensing Systems. MOOS-IvP home page. http://oceanai.mit.edu/moos-ivp/pmwiki/pmwiki.php?n=Main.HomePage, 2015. Accessed: October 2016.

[56] S. G. Lachmann and R. G. McKinstry. Software engineering of a navigation and guidane system for commercial aircraft. In *Proceedings of the AIAA Digital Avionics System Conference*, pages 1–15, Boston, US, 1975.

[57] G. Laporte, S. Ropke, and T. Vidal. Heuristics for the vehicle routing problem. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, pages 87–116. SIAM, Philadephia, 2nd edition, 2014.

[58] M. Lazar, W. Heemels, S. Weiland, and A. Bemporad. Stabilizing model predictive control of hybrid systems. *IEEE Transactions on Automatic Control*, 51(11):1813–1818, 2006.

[59] H. Li and Y. Shi. Robust distributed model predictive control of constrained continuous-time nonlinear systems: a robustness constraint approach. *IEEE Transactions on Automatic Control*, 59(6):1673–1678, 2014.

[60] L. Li, R. R. Negenborn, and B. De Schutter. Intermodal freight transport planning–a receding horizon control approach. *Transportation Research Part C: Emerging Technologies*, 60:77–95, 2015.

[61] S. Li, H. Zheng, R. R. Negenborn, and G. Lodewijks. Coordination for efficient transport over water. In *Proceedings of the 19th International Conference on Computer Supported Cooperative Work in Design*, pages 389–394, Calabria, Italy, 2015.

[62] Z. Li and J. Sun. Disturbance compensating model predictive control with application to ship heading control. *IEEE Transactions on Control Systems Technology*, 20(1): 257–265, 2012.

[63] Z. Li, J. Sun, and S. Oh. Path following for marine surface vessels with rudder and roll constraints: an MPC approach. In *Proceedings of American Control Conference*, pages 3611–3616, St. Louis, Missouri, 2009.

[64] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Efficient network-wide model-based predictive control for urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 24:122–140, 2012.

[65] X. Liu, Y. Shi, and D. Constantinescu. Distributed model predictive control of constrained weakly coupled nonlinear systems. *Systems & Control Letters*, 74:41–49, 2014.

[66] J. Lofberg. YALMIP: A toolbox for modeling and optimization in matlab. In *Proccedings of 2004 IEEE International Symposium on Computer Aided Control Systems Design*, pages 284–289, Taipei, Taiwan, 2004.

[67] J. Löfberg. Automatic robust convex programming. *Optimization Methods and Software*, 27(1):115–129, 2012.

[68] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves. Model predictive control for the operation of building cooling systems. *IEEE Transactions on Control Systems Technology*, 20(3):796–803, 2012.

[69] Maersk. Big ship means big pollution, right? http://www.maersk.com/en/test-section/wls-test/stories/big-ship-means-big-pollution-right, 2013. Accessed January 2016.

[70] J. Maestre, D. Muñoz De La Peña, and E. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32 (2):153–176, 2011.

[71] J. M. Maestre and R. R. Negenborn. *Distributed model predictive control made easy*. Springer, Dordrecht, 2014.

[72] J. Majohr and T. Buch. Modelling, simulation and control of an autonomous surface marine vehicle for surveying applications Measuring Dolphin MESSIN. In G. N. Roberts and R. Sutton, editors, *Advances in Unmanned Marine Vehicles*, volume 69, pages 329 – 352. MPG Books Ltd, Cornwall, UK, 2006.

[73] M. Maloni, J. A. Paul, and D. M. Gligor. Slow steaming impacts on ocean carriers and shippers. *Maritime Economics & Logistics*, 15(2):151–171, 2013.

[74] K. Margellos, P. Goulart, and J. Lygeros. On the road between robust optimization and the scenario approach for chance constrained optimization problems. *IEEE Transactions on Automatic Control*, 59(8):2258–2263, 2014.

[75] MATLAB. *R2014b*. The MathWorks Inc., Natick, Massachusetts, 2014.

[76] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[77] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[78] D. Q. Mayne, M. M. Seron, and S. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[79] N. Minorski. Directional stability of automatically steered bodies. *Journal of American Society of Naval Engineers*, 42(2):280–309, 1922.

[80] L. Moreira, T. I. Fossen, and C. G. Soares. Path following control system for a tanker ship model. *Ocean Engineering*, 34(14):2074–2085, 2007.

[81] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel. D-ADMM: A distributed algorithm for compressed sensing and other separable optimization problems. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 2869–2872, Kyoto, Japan, 2012.

[82] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel. Distributed ADMM for model predictive control and congestion control. In *Proceedings of 51st Annual Conference on Decision and Control*, pages 5110–5115, Maui, USA, 2012.

[83] T. Müller. *Automated guided vehicles*. IFS (Publications) Ltd./Springer-Verlag, UK/Berlin, 1983.

[84] W. Naeem, R. Sutton, J. Chudley, F. Dalgleish, and S. Tetlow. An online genetic algorithm based model predictive control autopilot design with experimental verification. *International Journal of Control*, 78(14):1076–1090, 2005.

[85] W. Naeem, T. Xu, R. Sutton, and A. Tiano. The design of a navigation, guidance, and control system for an unmanned surface vehicle for environmental monitoring. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 222(2):67–79, 2008.

[86] W. Naeem, G. W. Irwin, and A. Yang. COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics*, 22(6):669–678, 2012.

[87] NATO Industrial Advisory Group (NIAG). Pre-feasibility study on UAV autonomous operations. Technical report, NIAG Special Group 75, 2004.

[88] R. R. Negenborn. Delfia-1 autonomous vessel facility. http://www.negenborn.net/, 2016. Accessed: June 2016.

[89] R. R. Negenborn and J. M. Maestre. On 35 approaches for distributed MPC made easy. In J. M. Maestre and R. R. Negenborn, editors, *Distributed Model Predictive Control Made Easy*, pages 1–37. Springer, 2014.

[90] R. R. Negenborn, B. De Schutter, and J. Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21(3):353–366, 2008.

[91] L. A. Nguyen, M. D. Le, S. H. Nguyen, T. H. H. Nghiem, et al. A new and effective fuzzy PID autopilot for ships. In *SICE 2003 Annual Conference*, volume 3, pages 2647–2650, 2003.

[92] I. Norstad, K. Fagerholt, and G. Laporte. Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, 19(5): 853–865, 2011.

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

[94] F. Oldewurtel, C. N. Jones, A. Parisio, and M. Morari. Stochastic model predictive control for building climate control. *IEEE Transactions on Control Systems Technology*, 22(3):1198–1205, 2014.

[95] H. Y. Ong and J. C. Gerdes. Cooperative collision avoidance via proximal message passing. In *Proceedings of 2015 American Control Conference*, pages 4124–4130, Chicago, USA, 2015.

[96] K. W. Pang, Z. Xu, and C. L. Li. Ship routing problem with berthing time clash avoidance constraints. *International Journal of Production Economics*, 131(2):752–762, 2011.

[97] Parliament of the United Kingdom. International regulations for preventing collisions at sea. Technical report, IMO (The International Maritime Organisation), London, UK, 1972.

[98] A. Pavlov, H. Nordahl, and M. Breivik. MPC-based optimal path following for underactuated vessels. In *Proceedings of the 8th IFAC International Conference on Manoeuvring and Control of Marine Craft*, pages 340–345, Guaruj, Brazil, 2009.

[99] T. Perez, O. Smogeli, T. I. Fossen, and A. Sorensen. An overview of the marine systems simulator (MSS): a simulink toolbox for marine control systems. *Modeling, Identification and Control*, 27(4):259–275, 2006.

[100] Z. Pietrzykowski and J. Uriasz. The ship domain–a criterion of navigational safety assessment in an open sea area. *Journal of Navigation*, 62(01):93–108, 2009.

[101] Port of Rotterdam. Port Vision 2030. http://www.portofrotterdam.com/en/Port/port-in-general/port-vision-2030/Pages/default.aspx, 2011. Accessed: November 2014.

[102] Port of Rotterdam. Maasvlakte II. https://www.maasvlakte2.com/nl/index/, 2013. Accessed: May 2016.

[103] Port of Rotterdam. Expected weather conditions in the port of rotterdam, 2015. URL https://www.portofrotterdam.com/en/shipping/up-to-date-information/current-water-levels-flow-wind-and-visibility. Accessed: July 2016.

[104] J. A. Primbs and C. H. Sung. Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. *IEEE Transactions on Automatic Control*, 54(2):221–230, 2009.

[105] H. N. Psaraftis and C. A. Kontovas. Ship speed optimization: Concepts, models and combined speed-routing scenarios. *Transportation Research Part C: Emerging Technologies*, 44:52–69, 2014.

[106] H. N. Psaraftis, M. Wen, and C. A. Kontovas. Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31, 2016.

[107] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[108] Rafael Advanced Defense Systems Ltd. Unmanned naval patrol vehicle – PROTECTOR, December 2010. URL http://www.rafael.co.il/Marketing/351-1037-en/Marketing.aspx. Accessed: June 2016.

[109] J. B. Rawlings, D. Angeli, and C. N. Bates. Fundamentals of economic model predictive control. In *Proceedings of the 51st Annual Conference on Decision and Control*, pages 3851–3861, Maui, USA, 2012.

[110] J. B. Rawlings and D. Q. Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., Madison, USA, 2009.

[111] A. Richards and J. P. How. Robust distributed model predictive control. *International Journal of Control*, 80(9):1517–1531, 2007.

[112] M. P. Richter. *Analysis of operational manning requirements and deployment procedures for unmanned surface vehicles aboard US Navy ships*. PhD thesis, Naval Postgraduate School, Monterey, United States, 2006.

[113] S. Riverso and G. Ferrari-Trecate. Tube-based distributed control of linear constrained systems. *Automatica*, 48(11):2860–2865, 2012.

[114] G. Roussos and K. J. Kyriakopoulos. Decentralized navigation and conflict avoidance for aircraft in 3-d space. *IEEE Transactions on Control Systems Technology*, 20(6): 1622–1629, 2012.

[115] M. W. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29(1):17–29, 1995.

[116] R. Scattolini. Architectures for distributed and hierarchical model predictive control– a review. *Journal of Process Control*, 19(5):723–731, 2009.

[117] T. Schouwenaars, B. De Moor, E. Feron, and J. How. Mixed integer programming for multi-vehicle path planning. In *Proceedings of 6th European Control Conference*, pages 2603–2608, Porto, Portugal, 2001.

[118] H. J. Schroër, F. Corman, M. B. Duinkerken, R. R. Negenborn, and G. Lodewijks. Evaluation of inter terminal transport configurations at Rotterdam Maasvlakte using discrete event simulation. In *Proceedings of Winter Simulation Conference*, pages 1771–1782, Savannah, USA, 2014.

[119] M. S. Sedehi and R. Z. Farahani. An integrated approach to determine the block layout, agv flow path and the location of pick-up/delivery points in single-loop systems. *International Journal of Production Research*, 47(11):3041–3061, 2009.

[120] F. Semet, P. Toth, and D. Vigo. Classical exact algorithms for the capacitated vehicle routing problem. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, pages 37–57. SIAM, Philadephia, 2nd edition, 2014.

[121] R. Skjetne. *The maneuvering problem*. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2005.

[122] R. Skjetne, T. I. Fossen, and P. V. Kokotović. Adaptive maneuvering with experiments for a model ship in a marine control laboratory. *Automatica*, 41(2):289–298, 2005.

[123] SNAME: The Society of Naval Architects and Marine Engineers. Nomenclature for treating the motion of a submerged body through a fluid. *Technical and Research Bulletin*, pages 1–5, 1952.

[124] A. J. Sørensen. A survey of dynamic positioning control systems. *Annual Reviews in Control*, 35(1):123–136, 2011.

[125] T. H. Summers and J. Lygeros. Distributed model predictive consensus via the alternating direction method of multipliers. In *in Proceedings of 50th Annual Allerton Conference on Communication, Control, and Computing*, pages 79–84, Urbana IL, USA, 2012.

[126] R. Sutton. Design of the multi-role springer unmanned surface vehicle. Technical report, Plymouth University, Plymouth, UK, 2007.

[127] N. Tarcai, C. Virágh, D. Ábel, M. Nagy, P. L. Várkonyi, G. Vásárhelyi, and T. Vicsek. Patterns, transitions and the role of leaders in the collective dynamics of a simple robotic flock. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(04): 1–26, 2011.

[128] Terex Corporation. Terex Gottwald automated guided vehicles. http://www.terex.com/port-solutions/en/products/automated-guided-vehicles/index.htm, 2016. Accessed: May 2016.

[129] K. Tierney, S. Voß, and R. Stahlbock. A mathematical model of inter-terminal transportation. *European Journal of Operational Research*, 235(2):448–460, 2014.

[130] P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*, volume 18. SIAM, Philadelphia, 2nd edition, 2014.

[131] P. Trodden. Feasible parallel-update distributed MPC for uncertain linear systems sharing convex constraints. *Systems & Control Letters*, 74:98–107, 2014.

[132] P. Trodden and A. Richards. Distributed model predictive control of linear systems with persistent disturbances. *International Journal of Control*, 83(8):1653–1663, 2010.

[133] P. Trodden and A. Richards. Cooperative distributed MPC of linear systems with coupled constraints. *Automatica*, 49(2):479–487, 2013.

[134] R. Vilanova and A. Visioli. *PID Control in the Third Millennium*. Springer, Springer-Verlag London, UK, 2012.

[135] I. F. Vis. Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170(3):677–709, 2006.

[136] I. F. Vis, R. M. B. de Koster, and M. W. Savelsbergh. Minimum vehicle fleet size under time-window constraints at a container terminal. *Transportation Science*, 39 (2):249–260, 2005.

[137] A. Wahl and E. Gilles. Track-keeping on waterways using model predictive control. In *Proceedings of the IFAC Conference on Control Applications in Marine Systems*, pages 149–154, Fukuoka, Japan, 1998.

[138] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem. Cooperative car-following control: Distributed algorithm and impact on moving jam features. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1459–1471, 2016.

[139] X. Wang, Z. Zou, T. Li, and W. Luo. Path following control of underactuated ships based on nonswitch analytic model predictive control. *Journal of Control Theory and Applications*, 8(4):429–434, 2010.

[140] H. P. Williams. Logical problems and integer programming. *Bulletin of the Institute of Mathematics and its Applications*, 13:18–20, 1977.

[141] L. A. Wolsey. *Integer programming*, volume 42. Wiley, New York, 1998.

[142] J. Xin, R. R. Negenborn, and G. Lodewijks. Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control. *Transportation Research Part C: Emerging Technologies*, 44:214–230, 2014.

[143] J. Xin, R. R. Negenborn, F. Corman, and G. Lodewijks. Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance. *Transportation Research Part C: Emerging Technologies*, 60: 377–396, 2015.

[144] M. N. Zeilinger, M. Morari, and C. N. Jones. Soft constrained model predictive control with robust stability guarantees. *IEEE Transactions on Automatic Control*, 59(5):1190–1202, 2014.

[145] H. Zheng, Z. Huang, C. Wu, and R. R. Negenborn. Model predictive control for intelligent vehicle lane change. In *Proceedings of the 2nd Internation Conference on Transportation Information and Safety*, pages 265–276, Wuhan, China, 2013.

[146] H. Zheng, R. R. Negenborn, and G. Lodewijks. Trajectory tracking of autonomous vessels using model predictive control. In *Proceedings of the 19th IFAC World Congress*, pages 8812–8818, Cape Town, South Africa, August 2014.

[147] H. Zheng, R. R. Negenborn, and G. Lodewijks. Survey of approaches for improving the intelligence of marine surface vehicles. In *Proceedings of the 16th International Conference on Intelligent Transportation Systems*, pages 1217–1223, Den Haag, The Netherlands, 2013.

[148] H. Zheng, R. R. Negenborn, and G. Lodewijks. Cooperative distributed collision avoidance based on ADMM for waterborne AGVs. In F. Corman, S. Voß, and R. R. Negenborn, editors, *Computational Logistics*, pages 181–194. Cham, Switzerland, 2015.

[149] H. Zheng, R. R. Negenborn, and G. Lodewijks. Fast ADMM for distributed model predictive control of cooperative waterborne AGVs. *Accepted for publication in IEEE Transactions on Control Systems Technology*, In press, 2016. doi: http://dx.doi.org/10.1109/TCST.2016.2599485.

[150] H. Zheng, R. R. Negenborn, and G. Lodewijks. Predictive path following with arrival time awareness for waterborne AGVs. *Transportation Research Part C: Emerging Technologies*, 70:214 – 237, 2016.

[151] H. Zheng, R. R. Negenborn, and G. Lodewijks. Explicit use of probabilistic distributions in robust predictive control of waterborne AGVs - a cost-effective approach. In *Proceedings of the 15th European Control Conference (ECC16)*, pages 1278 – 1283, Aalborg, Denmark, 2016.

[152] H. Zheng, R. R. Negenborn, and G. Lodewijks. Robust distributed predictive control of waterborne AGVs – a cooperative and cost-effective approach. *Submitted to a journal*, 2016.

[153] H. Zheng, R. R. Negenborn, and G. Lodewijks. Closed-loop scheduling and control of waterborne AGVs for energy-efficient inter terminal transport. *Transportation Research Part E: Logistics and Transportation Review*, In press, 2016. doi: http://dx.doi.org/10.1016/j.tre.2016.07.010.

Glossary

List of symbols and notations

Below follows a list of the most frequently used symbols and notations in this thesis.

a_i	arrival time at node i
$\boldsymbol{a}$	binary vector parameterizing the distribution function
$\boldsymbol{a}^*$	LP-relaxed node solution
$\hat{\boldsymbol{a}}_p$	copied binary parameterizing vector by waterborne AGV p
$\boldsymbol{A}$	Jacobian state matrix
$\boldsymbol{A}_{\boldsymbol{K}_p}$	state matrix with feedback
$\mathcal{A}$	virtual arc set
b	environmental disturbances
$\boldsymbol{b}_\mathrm{p}$	binary switching logic decision variable
$\{b_p\}$	the body-fixed coordinate system of waterborne AGV p
$\boldsymbol{b}$	binary arrival time decision variable
$\boldsymbol{b}_{\mathrm{obs},i}, i = 1,2,3,4$	binary obstacle avoidance decision variable
$\bar{b}$	predicted disturbance force
B_p^j, B_q^j	outboundings of uncertainty perturbation positions of waterborne AGV p and q
$B_{u_p}^j$	outbounding of uncertain perturbation control inputs of waterborne AGV p
$B_{x_p}^j$	outbounding of uncertain perturbation states of waterborne AGV p
$\boldsymbol{B}$	Jacobian control input matrix
$\mathcal{B}$	berth set
$c_1, \cdots, c_6$	weight parameters
$\boldsymbol{C}$	Jacobian position matrix of waterborne AGV p
$\mathcal{C}_\mathrm{b}$	set of nodes representing the same physical berths
$\mathcal{C}_{x_p}$	convex set constraints on perturbation states of waterborne AGV p
$\mathcal{C}_{u_p}$	convex set constraints on perturbation control inputs of waterborne AGV p
$\mathcal{C}_{r_{p,q}}$	convex set constraints on perturbation positions variables of waterborne AGVs p and q
d_i	delivery berth of request i or delay time at node i
d_{ij}	distance between nodes i and j
d_s	safety distance from the obstacle
$\boldsymbol{D}$	Jacobian position matrix of waterborne AGV q

$D_{\mathbf{c}}$	communication range of waterborne AGVs
$D_{\mathbf{s}}$	minimum safety distance between waterborne AGVs
$\mathcal{D}_{\mathrm{n}}$	delivery node set
$e_{p,q}$	edge between waterborne AGVs p and q
$\boldsymbol{E}$	Jacobian disturbance matrix
$\mathcal{E}, \mathcal{E}_s$	set of couplings between waterborne AGVs
$\mathcal{G}$	waterborne AGV graphs
$\mathcal{G}_{\mathrm{s}}$	virtual scheduling graph
i	request or node ID
i^*	branch point by SOS1 branching
I_C	indicator function for $\mathcal{C}_{r_{p,q}}$
I_{ij}	binary variable for whether node i visited before node j, $i, j \in \mathcal{C}_b$
I_x, I_u	number of state and control input constraints
I	index set for $\boldsymbol{a}$
I'	subset of I
j	iteration number in ADMM
J^*	incumbent objective
J^*_{LP}	LP-relaxed node objective
$J^j_{\mathrm{P}}, J^j_{\mathrm{D}}$	primal and dual objectives at iteration j
k	discrete time step
$\boldsymbol{K}_p$	time-varying feedback gain of waterborne AGV p
l_j	length of path segment j
L	vector of berths corresponding to $\mathcal{N}$
$\mathcal{L}_\rho$	ρ-augmented Lagrangian function
m_{c}	mass of one TEU container
M	a big value
n, n_s	number of waterborne AGVs
n_u	number of speed levels
n_b	dimension of $\boldsymbol{a}$
$n_{\mathcal{G}}$	number of subgraphs
$\{n\}$	the inertial coordinate system
N	discrete preferable arrival times
$N_{\max}$	discrete latest arrival times
N_p	prediction horizon
$\mathcal{N}$	virtual node set
p	waterborne AGV ID or probability of $b \in [-z, z]$
p_i	pick-up berth of request i
$\{p_j\}$	the path coordinate j
$\boldsymbol{P}$	discrete probability vector

$\mathcal{P}_\mathrm{n}$	pick-up node set
q_i	container number of request i
$\boldsymbol{Q}$	state cost weight matrix
$\boldsymbol{Q}_f$	terminal state cost weight matrix
r	discrete speed levels
r_p	yaw rate of waterborne AGV p
r^j	primal residual at iteration j of ADMM
$\mathbf{r}_p$	position vector of waterborne AGV p
$\boldsymbol{R}$	control input weight matrix
$\mathcal{R}$	request set
s	subgraph ID
s^j	dual residual at iteration j of ADMM
s_i	service time of request i
s_v	traveled distance of waterborne AGV v on a route
S_{ij}	binary variable for whether node $i, j \in \mathcal{C}_b$ visited by different waterborne AGVs
t	continuous time
$t_{i,\min}$	release time of request i
$t_{i,\max}$	due time of request i
$\underline{t_\mathrm{f}}$	continuous and discrete preferable arrival times
$\overline{t_\mathrm{f}}$	continuous and discrete latest arrival times
T	time interval between service time slots of different waterborne AGVs at the same berth
T_s	sampling time
$\boldsymbol{T}$	diagonal translation matrix
u_{ij}	speed a waterborne AGV travels at on leg $i \to j$
u_p	surge speed of waterborne AGV p
$u_{r,\min}, u_{r,\max}$	minimum and maximum speeds of speed level r
$\bar{u}_r$	average speed of speed level r
$\boldsymbol{u}_p$	system control input vector of waterborne AGV p
$\Delta\bar{\boldsymbol{u}}_p$	nominal perturbation control inputs of waterborne AGV p
$\Delta\tilde{\boldsymbol{u}}_p$	uncertain perturbation control inputs of waterborne AGV p
$\boldsymbol{u}_{s,\mathrm{l}}$	lower level control input vector
$\boldsymbol{u}_{s,\mathrm{h}}$	higher level control input vector
$\boldsymbol{U}_{\mathcal{G}_s}$	control input vector of subgraph s
$\mathcal{U}_p$	control input tube of waterborne AGV p
v	waterborne AGV ID
v_p	sway speed of waterborne AGV p
$\boldsymbol{\nu}_p$	velocity vector of waterborne AGV p
$\boldsymbol{\nu}_{p,\mathbf{r}}$	relative velocity vector of waterborne AGV p with respect to current speed
$\boldsymbol{\nu}_{p_\min}$	minimum velocity vector of waterborne AGV p
$\boldsymbol{\nu}_{p_\max}$	maximum velocity vector of waterborne AGV p
$\boldsymbol{\nu}(t_\mathrm{f})$	final velocity vector

$V_{\mathbf{c}}$	current speed
$V_i, i = 1, 2, \cdots$	numbered waterborne AGVs
$\mathcal{V}, \mathcal{V}_s$	set of waterborne AGVs
$\mathcal{V}_{\mathrm{o}}, \mathcal{V}_{\mathrm{e}}$	start and end location sets of waterborne AGVs
w_i	waiting time at node i
$w_{\max}, d_{\max}$	maximum waiting and delay times at all nodes
$\mathcal{W}$	compact set for uncertainty bounds
x_{ijv}	binary variable for whether waterborne AGV v traveling from node $i \to j$
x_p	coordinate along X_n axis of waterborne AGV p
$\boldsymbol{x}_{s,\mathrm{l}}$	lower level state vector
$\boldsymbol{x}_{s,\mathrm{h}}$	higher level state vector
$\boldsymbol{x}_p$	system state vector of waterborne AGV p
$\boldsymbol{x}^e_{\mathrm{p}_j}$	tracking error vector in the path coordinate $\{p_j\}$
$\Delta\bar{\boldsymbol{x}}_p$	nominal perturbation states of waterborne AGV p
$\Delta\tilde{\boldsymbol{x}}_p$	uncertain perturbation states of waterborne AGV p
$\boldsymbol{X}_{\mathcal{G}_s}$	state vector of subgraph s
$\mathcal{X}_p$	state tube of waterborne AGV p
$\Delta\tilde{\mathcal{X}}_p$	convex sets of uncertain perturbation states of waterborne AGV p
y_p	coordinate along Y_n axis of waterborne AGV p
y_i	load on board waterborne AGVs upon arriving node i
z	uncertainty bounds
z_{iv}	binary variable for whether node i visited by waterborne AGV v
$\boldsymbol{Z}$	discrete bound vector
$\beta_{\mathbf{c}}$	current angle
δ_s^{j+1}	deviations of $\Delta\boldsymbol{r}_p^{j+1}$ from $\Delta\hat{\boldsymbol{r}}_p^{j}$ of subgraph s
$\varepsilon^{\mathrm{pri}}$	primal tolerance
$\varepsilon^{\mathrm{dual}}$	dual tolerance
$\boldsymbol{\eta}_p$	pose vector of waterborne AGV p
$\boldsymbol{\eta}_{p_j}$	pose vector in the path coordinate j
$\boldsymbol{\eta}(t_{\mathrm{f}})$	final pose vector
$\boldsymbol{\lambda}_{p,a}$	dual variable for consensus constraints on binary parametering vectors
$\boldsymbol{\lambda}_{p,r}$	dual variable for position variable consensus constraints of waterborne AGV p
$\tau_{u,p}$	surge force of waterborne AGV p
$\tau_{v,p}$	sway force of waterborne AGV p
$\tau_{r,p}$	yaw moment of waterborne AGV p
$\boldsymbol{\tau}$	control input vector of waterborne AGV p
$\boldsymbol{\tau}_{p_{\min}}$	minimum control input vector of waterborne AGV p
$\boldsymbol{\tau}_{p_{\max}}$	maximum control input vector of waterborne AGV p

ψ_p	heading angle of waterborne AGV p
ψ_j	angle of path segment j with X_n axis
Σ	covariance of disturbance prediction distribution

List of abbreviations

The following abbreviations are used in this thesis:

waterborne AGVs	waterborne Autonomous Guided Vessels
AGVs	Automated Guided Vehicles
ITT	Inter Terminal Transport
TEU	Twenty-foot Equivalent Unit
MPC	Model Predictive Control
NGC	navigation, guidance, and control
LOS	line-of-sight guidance
DOF	degree-of-freedom
PPF-ATA	predictive path following with arrival time awareness
MIQP	mixed integer quadratic programming
DPPF-ATA	distributed predictive path following with arrival time awareness
ADMM	the Alternating Direction Method of Multipliers
DMPC	distributed Model Predictive Control
RDMPC	robust distributed Model Predictive Control
B&B	branch and bound
LP	linear programming
MILP	mixed integer linear programming
GUB	generalized upper bound
SOS1	special ordered sets of type I
PDP	the pick-up and delivery problem
VRP	the vehicle routing problem

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 150 titles see the TRAIL website: www.rsTRAIL.nl.

The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Zheng, H., *Coordination of Waterborn AGVs*, T2016/25, December 2016, TRAIL Thesis Series, the Netherlands

Yuan, K., *Capacity Drop on Freeways: Traffic Dynamics, Theory and Modeling*, T2016/24, December 2016, TRAIL Thesis Series, the Netherlands

Li, S., *Coordinated Planning of Inland Vessels for Large Seaports*, T2016/23, December 2016, TRAIL Thesis Series, the Netherlands

Berg, M. van den, *The Influence of Herding on Departure Choice in Case of Evacuation: Design and Analysis of a Serious Gaming Experimental Set-up*, T2016/22, December 2016, TRAIL Thesis Series, the Netherlands

Luo, R., *Multi-Agent Control of Urban Transportation Networks and of Hybrid Systems with Limited Information Sharing*, T2016/21, November 2016, TRAIL Thesis Series, the Netherlands

Campanella, M., *Microscopic Modelling of Walking Behavior*, T2016/20, November 2016, TRAIL Thesis Series, the Netherlands

Horst, M. van der, *Coordination in Hinterland Chains: An Institutional Analysis of Port-related Transport*, T2016/19, November 2016, TRAIL Thesis Series, the Netherlands

Beukenkamp, *Securing Safety: Resilience Time as a Hidden Critical Factor*, T2016/18, October 2016, TRAIL Thesis Series, the Netherlands

Mingardo, G., *Articles on Parking Policy*, T2016/17, October 2016, TRAIL Thesis Series, the Netherlands

Duives, D.C., *Analysis and Modelling of Pedestrian Movement Dynamics at Large-scale Events*, T2016/16, October 2016, TRAIL Thesis Series, the Netherlands

Wan Ahmad, W.N.K., *Contextual Factors of Sustainable Supply Chain Management Practices in the Oil and Gas Industry*, T2016/15, September 2016, TRAIL Thesis Series, the Netherlands

Liu, X., *Prediction of Belt Conveyor Idler Performance*, T2016/14, September 2016, TRAIL

Thesis Series, the Netherlands

Gaast, J.P. van der, *Stochastic Models for Order Picking Systems*, T2016/13, September 2016, TRAIL Thesis Series, the Netherlands

Wagenaar, J.C., *Practice Oriented Algorithmic Disruption Management in Passenger Railways*, T2016/12, September 2016, TRAIL Thesis Series, the Netherlands

Psarra, I., *A Bounded Rationality Model of Short and Long-Term Dynamics of Activity-Travel Behavior*, T2016/11, June 2016, TRAIL Thesis Series, the Netherlands

Ma, Y., *The Use of Advanced Transportation Monitoring Data for Official Statistics*, T2016/10, June 2016, TRAIL Thesis Series, the Netherlands

Li, L., *Coordinated Model Predictive Control of Synchromodal Freight Transport Systems*, T2016/9, June 2016, TRAIL Thesis Series, the Netherlands

Vonk Noordegraaf, D.M., *Road Pricing Policy Implementation*, T2016/8, June 2016, TRAIL Thesis Series, the Netherlands

Liu, S., Modeling, *Robust and Distributed Model Predictive Control for Freeway Networks*, T2016/7, May 2016, TRAIL Thesis Series, the Netherlands

Calvert, S.C., *Stochastic Macroscopic Analysis and Modelling for Traffic Management*, T2016/6, May 2016, TRAIL Thesis Series, the Netherlands

Sparing, D., *Reliable Timetable Design for Railways and Connecting Public Transport Services*, T2016/5, May 2016, TRAIL Thesis Series, the Netherlands

Rasouli, S, *Uncertainty in Modeling Activity-Travel Demand in Complex Urban Systems*, T2016/4, March 2016, TRAIL Thesis Series, the Netherlands

Vries, J. de, *Behavioral Operations in Logistics*, T2016/3, February 2016, TRAIL Thesis Series, the Netherlands

Goñi-Ros, B., *Traffic Flow at Sags: Theory, Modeling and Control*, T2016/2, March 2016, TRAIL Thesis Series, the Netherlands

Khademi, E., *Effects of Pricing Strategies on Dynamic Repertoires of Activity-Travel Behaviour*, T2016/1, February 2016, TRAIL Thesis Series, the Netherlands

Cong, Z., *Efficient Optimization Methods for Freeway Management and Control*, T2015/17, November 2015, TRAIL Thesis Series, the Netherlands

Kersbergen, B., *Modeling and Control of Switching Max-Plus-Linear Systems: Rescheduling of railway traffic and changing gaits in legged locomotion*, T2015/16, October 2015, TRAIL Thesis Series, the Netherlands

Brands, T., *Multi-Objective Optimisation of Multimodal Passenger Transportation Networks*, T2015/15, October 2015, TRAIL Thesis Series, the Netherlands

Ardi, Ö., *Road Pricing Policy Process: The interplay between policy actors, the media and public*, T2015/14, September 2015, TRAIL Thesis Series, the Netherlands

Samenvatting

Coördinatie van Water AGVs

De verwachte grotere containeroverslag en de beperkte behandelingscapaciteit van bestaande haveninfrastructuren leggen een steeds grotere druk op havens om het concurrentievermogen te verbeteren. Binnen containerterminals worden aan de landzijde al decennia lang op grote schaal automatisch geleide voertuigen gebruikt om operationele efficiëntie en duurzaamheid te verbeteren. Vervoer tussen de terminals, zogenaamd Inter Terminal Transport (ITT), wordt op dit moment vooral gerealiseerd met behulp van bemande vrachtwagens. Het wegverkeer in havengebieden met beperkte wegcapaciteit is echter al zwaar. Voor geografisch complexe havens, zoals de haven van Rotterdam, kunnen reisafstanden tussen terminals over land veel langer zijn dan de afstanden tussen die terminals over water. Uitbreiding van de bestaande fysieke infrastructuur voor transport brengt extreem hoge kosten met zich mee. Als alternatief moeten meer efficiënte en duurzame manieren om havenlogistiek te verbeteren worden onderzocht.

Dit promotieonderzoek stelt een nieuw vervoerstype voor containers voor: Water AGVs ("Waterborne Autonomous Guided Vessels"), met name voor de het uitvoeren van ITT. De nadruk ligt in dit onderzoek op geavanceerde besturings- en planningsstrategieën voor het coördineren van Water AGVs. Besturingstechnieken gebaseerd op modelgebaseerd voorspellend regelen ("model predictive control" (MPC)) van gedetailleerde Water AGV dynamiek worden onderzocht en coördinatietechnieken voor het oplossen van planningsproblemen tussen meerdere Water AGVs voor het uitvoeren van ITT worden voorgesteld.

De waarde van Water AGVs wordt geanalyseerd voor vier belangrijke ITT scenario's met toenemende mate van complexiteit:

- Het individuele Water AGV scenario.

  In dit scenario wordt een enkele Water AGV een ITT taak toegewezen, bestaande uit het verplaatsen van containers op bepaalde tijden tussen gespecificeerde herkomst en bestemmingsterminals. De Water AGV streeft er hierbij naar om een referentiepad nauwkeurig te volgen en op het juiste moment, op een energie-efficiënte manier en zonder aanvaringen aan te komen op de bestemming. Om deze doelen te bereiken wordt een besturingstechniek voor een Water AGV voorgesteld gebaseerd op het idee van voorspellend padvolgen met aankomsttijd bewustzijn ("predictive path following with arrival time awareness" (PPF-ATA)). De PPF-ATA besturing is gebaseerd op een zogenaamde aaneengesloten-pad assenstelsel, een referentieschakellogica, en een twee-level dubbele-integrator schema voor referentiepadparameterisatie. Simula-

tieresultaten van relevante ITT case studies in de haven van Rotterdam laten zien dat Water AGVs de toegewezen ITT taken succesvol uit kunnen voeren met de voorgestelde aanpak.

- Meerdere coöperatieve Water AGVs.

  In het scenario met meerdere coöperatieve Water AGVs wordt elke Water AGV een ITT taak toegewezen en moeten Water AGVs een veilige afstand van elkaar houden. Bij de uitvoering van ITT taken lossen de Water AGVs lokale problemen in parallel op, terwijl zij een overkoepelende doelstelling nastreven en antibotsingsbeperkingen respecteren. De betrokken Water AGVs worden hierbij gerepresenteerd met tijdsvariërende grafen. Een parallelle coöperatieve controller ("distributed PPF-ATA") wordt voorgesteld om acties te bepalen op basis van gedistribueerde MPC en de wisselende richting van multipliers methode (ADMM). Een variant, snelle ADMM ("fast ADMM") wordt voorgesteld om mogelijk trage convergentie van conventionele ADMM te verbeteren. Simulatieresultaten tonen aan dat alle Water AGVs succesvol en veilig hun toegewezen ITT taken uitvoeren met de voorgestelde besturing.

- Meerdere Water AGVs in een omgeving met onzekerheid.

  Wanneer er onzekerheden bestaan in omgevingen met verstoringen moeten prestaties en veiligheid van Water AGVs worden gerealiseerd op een robuuste gedistribueerde manier. Voor onzekerheden met stochastische eigenschappen en met oneindige basis, is het onmogelijk en onnodig om tegen alle mogelijke onzekerheidsrealisaties bestand te zijn. Het concept van kosteneffectieve robuustheid wordt voorgesteld en de trade-off tussen de waarschijnlijkheid van robuustbare onzekerheden en prestaties van het systeem worden geoptimaliseerd met een robuust gedistribueerd MPC (RDMPC) raamwerk.

  Naast het voorkomen van botsingen voor meerdere Water AGVs zoals in het deterministische geval moet overeenstemming worden bereikt over het niveau van overkoepelende systeemrobuustheid wanneer er onzekerheden bestaan. Een efficiënte geïntegreerde "branch and bound" (B&B) methode en het ADMM algoritme worden voorgesteld om het RDMPC probleem op te lossen. Het algoritme maakt gebruik van speciale gesorteerde waarschijnlijkheidssets die slimme zoekopdrachten in de B&B doen en integreert vertakkingscriteria met tussentijdse ADMM resultaten voor vroegtijdige beëindiging. Simulatieresultaten tonen aan dat, ondanks de onzekerheden, alle Water AGVs nog steeds de toegewezen ITT taken succesvol en veilig uitvoeren. De trade-off tussen het robuustheidsniveau en de prijs van robuustheid worden systematisch geoptimaliseerd voor meerdere Water AGVs met onzekerheden.

- Gesloten-lus planning en besturing van Water AGVs voor een autonomoon ITT systeem.

  Gegeven een set van ITT aanvragen en een vloot van Water AGVs is een gesloten-lus plannings- en besturingsaanpak voorgesteld om beslissingen voor het ITT-systeem op een autonome manier te maken. Gesloten-lus betekent hierbij dat zowel planning en besturingsproblemen continu worden opgelost. Op een hiërarchische manier omvat het planningsprobleem het bepalen van de volgorde en de tijden waarop terminals worden bezocht voor het laden/lossen van voorgeschreven hoeveelheden containers

door elke Water AGV. Het besturingsprobleem omvat het op een coöperatieve gedistribueerde manier bepalen van de optimale acties die genomen moeten worden om de gedetailleerde Water AGV dynamiek de toegewezen plannen uit te laten voeren. Daarnaast beschouwt het planningsprobleem gecoördineerd afmeren rekening houdend met de benodigde minimale tijdsintervallen tussen afmeren van verschillende Water AGVs op dezelfde ligplaats. De gesloten-lus planning en besturingsbenadering maakt het mogelijk dat lagere niveau factoren zoals onvoorziene fysieke systeembeperkingen, verstoringen, en de moeilijke botsingsvermijding kunnen worden gentegreerd in tijdig vernieuwde hogere niveau plannen. Simulatieresultaten tonen aan dat energie-efficiënte plannen worden gegenereerd die voldoen aan de plannings- en besturingsbeperkingen van de Water AGVs. Gecoördineerde tijden voor het afmeren worden ook bereikt voor alle Water AGV ligplaatsen. Daarnaast is de zogenaamde "non-performance" lager met behulp van de voorgestelde gesloten-lus aanpak dan een open-lus aanpak. Het voorgestelde autonome ITT systeem kan met behulp van Water AGVs een effectieve bijdrage aan slimmere havenlogistiek leveren.

In het kort, dit proefschrift stelt Water AGVs voor als een innovatieve en effectieve manier voor het uitvoeren van ITT. Hoewel de voorgestelde besturing- en planningsmethodes zijn ontworpen voor Water AGVs en zijn geoptimaliseerd voor ITT, kan hun toepassing worden gegeneraliseerd naar andere soortgelijke coördinatiescenarios.

Summary

Coordination of waterborne AGVs

The possible larger amount of container throughput and the limited handling capacities of existing infrastructures impose increasingly high pressure on large ports in improving competitiveness. Inside container terminals, land-side automated guided vehicles have been used extensively for decades to improve terminal operational efficiency and sustainability. Transport between terminals, i.e., Inter Terminal Transport (ITT), is currently mainly realized by manned trucks. However, road traffic has already been heavy in port areas with limited land. For geographically complex ports, e.g., the port of Rotterdam, travel distances by land can be much longer than by water between some terminals. Expanding the existing physical transportation infrastructure might be an option, at extremely high costs nonetheless. As an alternative, more efficient and sustainable ways for port logistics need to be investigated.

This PhD research proposes a new type of container transporter, waterborne Autonomous Guided Vessels (waterborne AGVs), for ITT. The focus is on advanced control and scheduling strategies for coordinating waterborne AGVs. Particularly, control problems based on model predictive control (MPC) considering detailed waterborne AGV dynamics and scheduling problems of multiple waterborne AGVs given ITT requests are investigated comprehensively. Four representative ITT scenarios with increasing levels of complexity using waterborne AGVs are studied:

- *A single waterborne AGV scenario.*

  In this scenario, a waterborne AGV is assigned an ITT task to move containers between specified origin and destination terminals at specified times. Performance criteria are that the waterborne AGV needs to track a reference path smoothly and arrive at the destination punctually in an energy-efficient way. To achieve these goals, a predictive path following with arrival time awareness (PPF-ATA) controller for one waterborne AGV is proposed. The PPF-ATA controller consists of a connected path coordinate system in which the inertial waterborne AGV kinematics are remodeled, a reference switching logic to avoid overshoots, and a two-level double-integrator scheme for parameterizing reference paths. Moreover, obstacle avoidance can also be addressed by this controller. Simulation results of relevant ITT case studies in the port of Rotterdam show that waterborne AGVs accomplish assigned ITT tasks successfully according to the defined performance criteria with the proposed approach.

- *Multiple cooperative waterborne AGVs.*

  For multiple cooperative waterborne AGVs, each waterborne AGV is assigned an ITT task and waterborne AGVs should keep a safe distance away from others. While carrying out ITT tasks, waterborne AGVs solve local problems in parallel while minimizing an overall objective and satisfying coupling collision avoidance constraints. The involved waterborne AGVs are modeled as time-varying graphs and a parallel cooperative controller, distributed PPF-ATA, is proposed based on distributed MPC and the alternating direction method of multipliers (ADMM). Collision avoidance constraints are satisfied in the framework of ADMM via iterative decomposition and coordination. Furthermore, a variant, fast ADMM, is proposed to improve the possible poor convergence rates of the conventional ADMM. Both the ADMM and fast ADMM based controllers achieve cooperative distributed computations through iterations. Simulation results show that all waterborne AGVs successfully and safely accomplish their assigned ITT tasks with local decision making.

- *Multiple waterborne AGVs facing uncertain environmental disturbances.*

  When there exist uncertainties in environmental disturbances, performance and safety of waterborne AGVs still need to be achieved in a robust distributed way. For uncertainties with stochastic characteristics and with infinite support, being robust to all possible realizations is impossible and not necessary. The concept of cost-effective robustness is proposed and the trade-off between the probability of robustable uncertainties and system performance is optimized in a cost-effective robust distributed MPC (RDMPC) framework. Besides the agreement on collision avoidance couplings for multiple waterborne AGVs as in the deterministic case, further agreement on an overall system robustness level has to be reached when uncertainties exist. An efficient integrated branch and bound (B&B) and ADMM algorithm is proposed to solve the cost-effective RDMPC problem. The algorithm exploits the special ordered probability sets conducting smart search in B&B and integrates branching criteria with intermediate ADMM results for early termination. Simulation results show that, in spite of uncertainties, all waterborne AGVs still fulfill assigned ITT tasks successfully and safely. Cooperative distributed decision making is also iteratively achieved following ADMM. The trade-off between robustness level and the price of being robust is optimized systematically for multiple waterborne AGVs with uncertainties.

- *Closed-loop scheduling and control of waterborne AGVs for an autonomous ITT system.*

  Given a set of ITT requests and a fleet of waterborne AGVs, a closed-loop scheduling and control approach is proposed to make decisions for the ITT system in an autonomous way. Closed-loop means that both scheduling and control problems are solved in real-time. In a hierarchical way, the scheduling problem decides, for each waterborne AGV, the sequence of terminals to visit at specified times with loading/unloading specified amount of containers. The control problem decides, in a cooperative distributed way, the optimal forces and moment considering detailed waterborne AGV dynamics in order to carry out the assigned schedules. Moreover, the scheduling problem considers coordinated berthing with necessary time intervals between different waterborne AGVs at the same berth. The real-time closed-loop

scheduling and control approach enables that lower level factors such as unconsidered physical system limits, disturbances, and collision avoidance that are difficult, if not impossible, to be integrated in a scheduling problem can be reflected timely in the online updated schedules. Simulation results show that energy-efficient schedules are generated and fulfilled successfully by waterborne AGVs satisfying both scheduling and control constraints. Coordinated berthing time slots are also achieved for all waterborne AGV berths. The so-called "non-performance" rate is lower using the proposed closed-loop approach than an open-loop approach. The proposed autonomous ITT system using waterborne AGVs is demonstrated to be effective contributing to smarter port logistics.

Briefly, this PhD thesis proposes an innovative and effective way, waterborne AGVs, for conducting ITT. Although the proposed control and scheduling methodologies are designed for waterborne AGVs with the specifications of ITT, their applications can also be generalized to other similar coordination scenarios.

Curriculum vitae

Huarong Zheng was born in December, 1988 in Jingshan, Hubei, China. She obtained the B.Sc degree on Logistics Engineering in 2011 and the M.Sc degree on Intelligent Transportation Engineering in 2013, both from Wuhan University of Technology in Wuhan, China.

Starting from 2013, Huarong is sponsored by China Scholarship Council as a PhD candidate at the Department of Maritime and Transport Technology, Delft University of Technology, Delft, the Netherlands. In her PhD project, Huarong proposes advanced coordination strategies for a new type of container transport vehicles, waterborne AGVs, for Inter Terminal Transport. Her research interests include model predictive control, intelligent transportation systems, and autonomous vehicles.

List of my publications during PhD

1. H. Zheng, R.R. Negenborn, G. Lodewijks. Closed-loop scheduling and control of waterborne AGVs for energy-efficient Inter Terminal Transport. *Transportation Research Part E: Logistics and Transportation Review*, In press, 2016, http://dx.doi.org/10.1016/j.tre.2016.07.010.

2. H. Zheng, R.R. Negenborn, G. Lodewijks. Fast ADMM for Distributed Model Predictive Control of Cooperative Waterborne AGVs. *IEEE Transactions on Control Systems Technology*, 2016. In press, doi: http://dx.doi.org/10.1109/TCST.2016.2599485.

3. H. Zheng, R.R. Negenborn, G. Lodewijks. Predictive path following with arrival time awareness for waterborne AGVs. *Transportation Research Part C: Emerging Technologies*, vol. 70, pp. 214 – 237, 2016. http://dx.doi.org/10.1016/j.trc.2015.11.004.

4. H. Zheng, R.R. Negenborn, G. Lodewijks. Explicit use of probabilistic distributions in robust predictive control of waterborne AGVs - a cost-effective approach. In proceedings of *the 15th European Control Conference (ECC16)*, Aalborg, Denmark, pp. 1278 - 1283, June - July 2016.

5. H. Zheng, R.R. Negenborn, G. Lodewijks. Cooperative distributed collision avoidance based on ADMM for waterborne AGVs. In *Proceedings of the 6th International Conference on Computational Logistics (ICCL'15)*, Delft, The Netherlands, pp. 181 - 194, September 2015.

6. C. Liu, H. Zheng, R.R. Negenborn, X. Chu, L. Wang. Trajectory tracking control for underactuated surface vessels based on nonlinear Model Predictive Control. In *Proceedings of the 6th International Conference on Computational Logistics (ICCL'15)*, Delft, The Netherlands, pp. 160 - 180, September 2015.

7. S. Li, H. Zheng, R.R. Negenborn, G. Lodewijks. Coordination for efficient transport over water. In *Proceedings of the 19th IEEE International Conference on Computer Supported Cooperative Work in Design (IEEE CSCWD'15)*, Calabria, Italy, pp. 389 - 394, May 2015.

8. H. Zheng, R.R. Negenborn, G. Lodewijks. Model predictive control of a waterborne AGV at the operational level. In *Proceedings of the International Maritime and Port Technology and Development Conference (MTEC 2014)*, Trondheim, Norway, pp. 99 - 108, October 2014.

9. H. Zheng, R.R. Negenborn, G. Lodewijks. Trajectory tracking of autonomous vessels using model predictive control. In *Proceedings of the 19th IFAC World Congress (IFAC WC'14)*, Cape Town, South Africa, pp. 8812 - 8818, August 2014.

10. C. Liu, R.R. Negenborn, F. Ma, X. Chu, H. Zheng. Towards an Experimental Platform for Inland Waterway Ship Navigation. In *Proceedings of the 2014 IEEE International Conference on Networking, Sensing, and Control (ICNSC 2014)*, Miami, Florida, pp. 707 - 712, April 2014.

11. H. Zheng, R.R. Negenborn, G. Lodewijks. Survey of approaches for improving the intelligence of marine surface vehicles. In *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (IEEE ITSC 2013)*, The Hague, The Netherlands, pp. 1217 - 1223, October 2013.

12. H. Zheng, R.R. Negenborn, G. Lodewijks. Robust distributed predictive control of waterborne AGVs - a cooperative and cost-effective approach. Submitted to a journal, 2016.