Control and Coordination for Automated Container Terminals

Jianbin Xin

Cover photo: Courtesy of Maxime Felder

Control and Coordination for Automated Container Terminals

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 21 september 2015 om 12:30 uur
door

Jianbin XIN

Master of Science in Control Engineering, Xi'an Jiaotong University
geboren te Zhengzhou, Henan, 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	chairperson
Prof. dr. ir. G. Lodewijks	promotor
Dr. R.R. Negenborn	copromotor

Independent members:

Prof. dr. L.A. Tavasszy	Technology, Policy and Management, TU Delft
Prof. ir. T. Vellinga	Civil Engineering and Geosciences, TU Delft
Prof. dr. ir. J. Hellendoorn	Mechanical, Maritime and Materials, TU Delft
Prof. dr. H. Geerlings	Erasmus University Rotterdam
Prof. dr. ing. S. Voss	University of Hamburg

This dissertation has been completed in partial fulfillment of the requirements of the Dutch Institute of Systems and Control for graduate studies. The research described in this dissertation is supported by the Chinese Scholarship Council under grant 2010628012 and the VENI project "Intelligent multi-agent control for flexible coordination of transport hubs" (project 11210).

TRAIL Thesis Series T2015/13, the Netherlands TRAIL Research School
P.O. Box 5017
2600 GA Delft, the Netherlands
Phone: +31 (0) 15 278 6046
Email: info@rsTRAIL.nl

Published and distributed by: Jianbin Xin
E-mail: jianbinxin@outlook.com

ISBN 978-90-5584-194-3

Keywords: automated container terminals, hybrid systems, energy efficiency, free-ranging AGVs.

Printed in the Netherlands

To my parents

Preface

Eventually, throughout four-year research, I am able to summarize the results obtained from my PhD project and write this dissertation. Before starting this project, I was kicked out of another department because I cannot demonstrate sufficient capability to conduct independent research, and I was afraid to fail again since I have a kind of one-way ticket sponsored by the Chinese Scholarship Council for carrying out four-year PhD research. At that moment, one year had gone already.

Fortunately, I found an interesting topic which perfectly fits both my background (control theory) and my personal interest in transportation, motivating me to work on this interesting topic continuously and complete it. Still, it is not easy because I have to demonstrate something both new and beneficial. I read quite a number of papers, got inspired by other people's work, programed and debugged continuously, wrote a paper and revised over and over again, attended the first international conference and got the first journal paper published. From a collection of these small steps, now, I can clearly see my personal developments, and certainly I would like to thank these people who gave me support and help to complete this journey.

Above all, I gratefully acknowledge the grant from the China Scholarship Council and the support from the Faculty of Mechanical, Maritime and Materials, Delft University of Technology for supporting me during the developments of this thesis.

Secondly, I would like to thank my promotor Prof. Gabriël Lodewijks for providing the opportunity to work in the section of Transport Engineering and Logistics at the department of Maritime and Transportation Technology. Within this section, I really enjoy the atmosphere of solving problems in the real world. Although he has a very busy schedule, he can always give valuable comments and suggestions on my research. In particular, I appreciate his effort for speeding up the procedures needed for the defense.

Furthermore, I am sincerely grateful to my daily supervisor Dr. Rudy Negenborn. Under his patient supervision and assistance, I have developed skills for carrying out a scientific research project independently. His broad view on different applications has great influence on my scientific thinking. Special thanks go to him for providing the Latex layout and translating the summary in Dutch, as well as the provided subsidy for the last year.

In addition, I am thankful to all the people and companies involved in this project. I thank all participants of the VENI meeting and all anonymous reviewers for improving quality of the work. I also thank company TBA for sharing its knowledge. Special thanks go to Dr. F. Corman for his critical comments and informative discussions with respect to our joint work.

Moreover, I would like to thank all the colleagues in the section of Transport Engineering and Logistics. The food experience in Aula and evening working hours with these

colleagues is definitely unforgettable. Many thanks also go to my friends in the Netherlands and in China. Special thanks must be made to Prof. Guimin Chen, Dr. Hui Cao and Prof. Gang Cheng, who encouraged me to complete this project and gave constructive suggestions on my career.

And finally, but not least, I should thank my parents for their endless and unconditional support. With this support, I can go further than it is supposed to be.

Dank u wel.

Jianbin Xin,
Delft, August, 2015.

Contents

Chapter 1

Introduction

1.1 Container transport

Over the last decades, there has been a considerable growth in container transport globally. Using a container, the freight can be stored in a standardized steel box during the process of transport without being opened. The standardization results in flexibility, low transport cost and rapid transshipment [85], in particular when the cargo is transported over long distances. Due to these advantages, containers have been widely used for global freight transport. Fig.1.1 illustrates the growth of container transport in the last few years. A further increase of container transport is expected, as projected for the coming years.

The global container transport consists of an extremely large and complex arrangement of distribution networks and business activities. In the network a container is typically transported in an intermodal way [20], using a sequence of at least two transport modes (e.g., vessel, barge, train and truck) from its origin to its destination. The transshipment of containers involves manufactures, freight forwarders, shipping lines, terminal operators and customers, forming a large supply chain.

As the transport hub, container terminals play an important role in the container transport network. A container terminal represents the interface between different transport modalities (e.g., vessel, barge, train and truck). The transfer of containers from one transport mode to another is performed at an intermodal container terminal. As a result, the performance of container terminals influences container transport considerably.

Automated container terminals can improve handling capacity significantly and reduce investment cost [63, 65, 66]. The significant increase in handling performance of container terminals is attributed to advanced automated equipment (e.g., automated guided vehicles, automated stacking cranes). Currently several automated container terminals exist (e.g., ECT Delta, ECT Euromax, and RWG in Rotterdam and HHLA in Hamburg), while new ones are being built (e.g., APM terminal MV2 in Rotterdam). Automation has become the trend of container terminals for the future in the Western world.

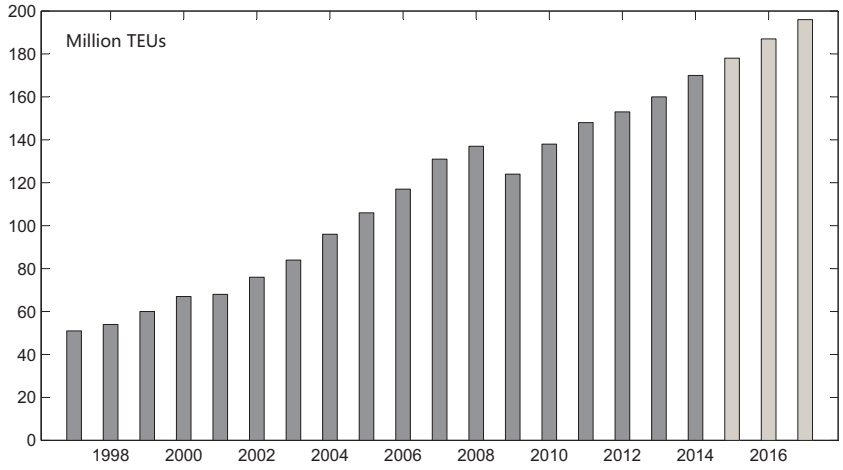


Figure 1.1: Global container trade including future projection, 1997-2017 (Based on data from Drewry Shipping Consultants [28], Clarkson Research Services [19], and IHS Global Insight [52].)

1.2 Problem statement

The terminal operation can typically be split in three parts: the sea or quayside operation, the stacking operation and the landside operation. The quayside operation is a major part of automated container terminals. At the quayside, vessels, in particular deep-sea vessels, arrive at and depart from the terminal with a great number of containers. The containers at automated container terminals are processed by a large number of unmanned machines (e.g., quay crane (QCs), automated guided vehicle (AGVs) and automated stacking cranes (ASCs)). For instance, the ECT Delta terminal at Rotterdam owns 36 QCs, 265 AGVs and 137 ASCs [32]. These unmanned machines are working in an interactive way and transport containers between the quayside area and the stacking area. These interactive quayside operations can be seen in Fig. 1.2.

The growing amount of containers that arrive and depart with container ships increases the pressure on terminal operators. In the year 2000, the capacity of a container vessel was typically 6,000-8,000 twenty-foot equivalent units (TEUs); in 2013, the number of containers carried by a container vessel can be up to 18,000 TEU [85]. Due to the increased size of a container vessel, the turnaround time of the container vessel may increase significantly if no measures are taken by terminal operators. This is against the shipping company's expectation. To retain the terminal's competitiveness, maintaining an acceptable turnaround time motivates terminal operators to improve the performance.

For improving the performance of the terminal, currently there are two major problems that must be addressed. The first problem is the energy efficiency of whole terminal operation, as raised for terminal operators due to the increased energy price and environmental stress [25]. Container terminals consume a great deal of energy that leads to a significant

Figure 1.2: The Euromax container terminal (Courtesy of ECT).

amount of CO_2. As an example of the order of magnitude, the yearly electricity consumption of the ECT Delta terminal in Rotterdam is around 45,000MWh with a yearly transshipment volume of 4,260,000 TEU, producing 71.3 kton CO_2 [99]. The balance between the handling capacity and energy consumption becomes a practical problem for terminal operators, since ports are expecting more sustainable container terminals by improving emission caps.

The second problem is the challenge for implementing more autonomous equipment at container terminals in order to improve operation efficiency. Soon, new developed GPS-based AGVs are expected to enter the market [15]. This new AGV allows free-ranging behaviour and shortens the driving distance considerably compared to the fixed path guided using makers, wires, lasers or computer vision. Nevertheless, the free-ranging behavior of AGVs increases the complexity for controlling terminal operations. On the one hand collision avoidance of two AGVs must be considered for safety reasons. On the other hand AGVs cooperate with other types of machines (e.g., QCs and ASCs) interactively for loading or unloading vessels. Therefore, advanced control algorithms for integrating the interaction of multiple AGVs and the interaction of AGVs with other types of equipment must be developed for increasing the terminal efficiency.

The research on container terminals that has been investigated intensively over the last decades [15, 93, 94], is mainly carried out in the field of the operations research. The results of the existing literature lead to positive advancements and provides the insights on improving the performance of container terminals, e.g., scheduling problems of interacting

machines for improving handling capacity of the quayside, dispatching of automated guided vehicles for reducing the cost for terminals operators.

Despite the accumulation of literature on container terminals, the two problems mentioned are not addressed completely. The knowledge of operations research focuses on the system level neglecting important individual properties of machines, which influence energy efficiency and detailed applicable scheduling when it comes to the collision avoidance of free-ranging AGVs. Therefore, there are several open issues related to the control of automated container terminals at the operational level:

- Energy management in seaports has not been investigated sufficiently [1]. Energy efficiency of container terminals has been addressed at the strategic level [84, 105]. Nevertheless, it is under development at the operational level.
- The way to implement free-ranging AGVs for performing a high handling capacity at container terminals is still not clear. The current scheduling scheme [17, 18, 59] cannot incorporate the detailed collision-free trajectory of AGVs. The interference between AGVs is difficult to anticipate without scheduling and controlling detailed movements of AGVs [59].

1.3 Research question

Following the scientific problems, this thesis aims to investigate *how to improve energy efficiency and implement autonomously moving equipment of automated container terminals at the operational level*.

This main research question leads to three key research questions:

1. To what extent can the energy consumption be reduced while maintaining an acceptable operational performance?
2. What complexity of control algorithms should be considered?
3. How can the collision-free trajectory planning of AGVs and other equipment be integrated with the scheduling of interacting machines in automated container terminals?

Before answering these three key research questions, an intensive literature review will be carried out to further motivate the choice for these research directions.

For answering these key questions, as a whole this thesis proposes a mathematical approach structured around the system and control framework. This framework is illustrated in Fig. 1.3. In this framework a container terminal is regarded as a collection of several subsystems in which each subsystem has its own dynamics and interacts with other subsystems. For controlling such a complex system, results of control theory, like model predictive control [82] and distributed control [92] are considered as the supportive tools ultimately improving the performance of container terminals.

1.4 Structure of this thesis

To answer the identified research questions of this thesis, in the following chapters these two major problems in operational control of automated container terminals are discussed and

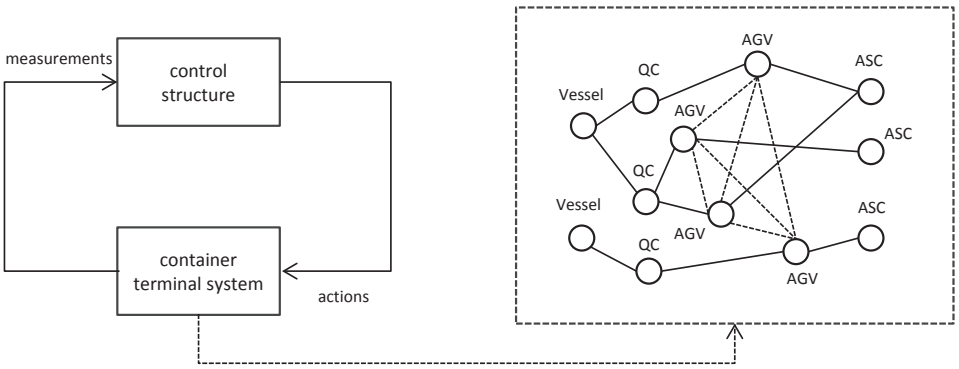


Figure 1.3: The system and control framework of a container terminal. (The solid line describes the interaction of two types of equipment for handling a container and the dash line indicates the possible collision between two AGVs.)

new solutions are proposed. Fig. 1.4 illustrates a grouping of the chapters in related subjects and an ordering in which the chapters can be read. This thesis is organized as follows:

- **Chapter 2** presents the background material and literature review on operational control of automated container terminals briefly. A description of the main characteristics of terminal operations is given. The existing approaches for operational control of container terminals are discussed. Moreover, the benchmark systems used later on in this thesis for analysis are proposed.

- In **Chapter 3** the energy efficiency for a compact container handling system is studied. The dynamics of inter-connected components are modeled using hybrid automata. After translating hybrid automata into mixed logic dynamical models, a hybrid model predictive controller (MPC) is proposed for achieving energy efficiency in real-time operation. This chapter partially answers key research question 1.

- In **Chapter 4** the topic emphasized is energy-efficiency of a medium-scale container terminal system. The case of the open-loop control problem is discussed in this chapter. The dynamics of container terminals are decomposed into discrete-event dynamics and continuous-time dynamics. Correspondingly a hierarchical control architecture for reducing control complexity of container terminals is proposed. This chapter partially answers key research question 1 and 2.

- **Chapter 5** follows up on the result of Chapter 4 and further explores energy-efficiency for real-time operations, still in the scope of the medium-scale terminal. Based on the result of the open-loop controller, an event-driven receding horizon controller is proposed for the closed-loop case. The proposed controller can reduce the computational burden and handle two types of uncertainties. This chapter partially answers key research question 1 and 2.

- In **Chapter 6** a large-scale terminal system is investigated. The research problem focuses on integration of the collision-free trajectory planning of free-ranging AGVs

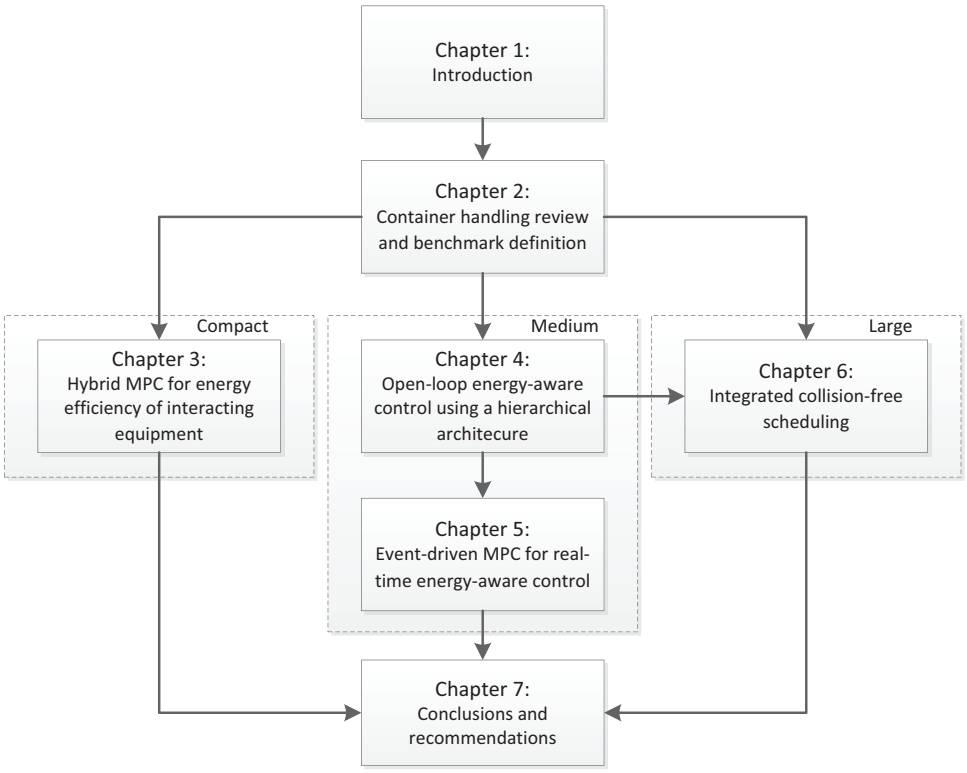


Figure 1.4: Road map. Arrows indicate read before relations.

in interaction with the other equipment with the scheduling of interacting machines. A 2-degree of freedom model for AGVs is considered, as well as static and dynamical obstacles. A new algorithm is proposed for generating the collision-free scheduling by solving a collection of small scale optimization problems. This chapter partially answers key research question 2 and 3.

- **Chapter 7** summarizes the results of this thesis and outlines directions for future research.

Chapter 2

Container handling review and benchmark definition

This chapter presents the overview of container handling and defines the benchmarks used throughout this thesis. Section 2.1 introduces container terminal operations. The research on decision problems of container terminals and advanced control technologies for large-scale systems are subsequently reviewed in Section 2.2 and Section 2.3. Section 2.4 proposes the details of three benchmark systems that will be used for the rest of the thesis.

2.1 Container terminals

A container terminal represents the interface among different transport modalities in an intermodal transport network. The container terminal typically connects the modalities of vessel, barge, train and truck. The transfer from one transport modality to another for containers is performed at the container terminal.

Container terminals handle two types of containers: inbound and outbound containers. Inbound containers are shifted from container vessels and are delivered to customers on land via railways, trucks or barges. Outbound containers are the opposite of inbound containers. The containers from railways, trucks or barges are transported to container vessels. In this thesis the transshipment containers between the barge and the vessel are considered to be stored temporarily in the stack.

The overview of a typical intermodal container terminal system is given in Fig. 2.1, which illustrates the handling areas and the equipment employed visually. An intermodal container terminal basically consists of the areas as shown in Fig. 2.2:

- Quayside area

  In the quayside area, vessels are located at the berth for loading or unloading containers using quay cranes (e.g., single or dual trolley cranes).

- Stacking area

  The stacking area is considered as a place for temporary storage of containers that are potentially shifted from one transport mode to another. Container stacking is either

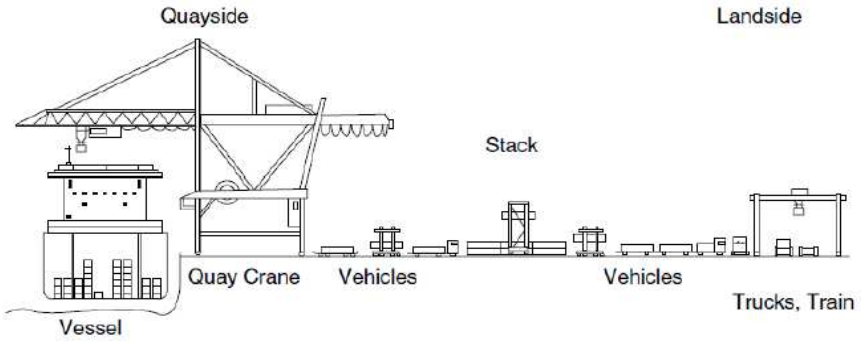


Figure 2.1: Overview of a typical container terminal system [94].

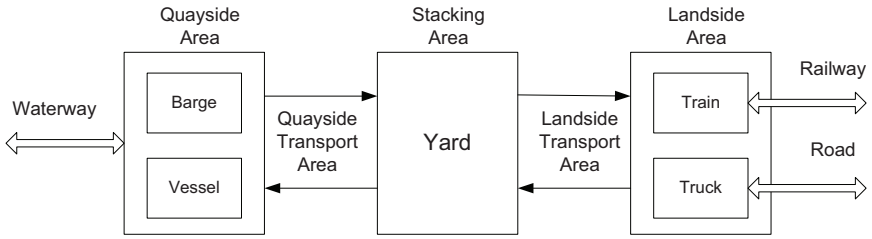


Figure 2.2: The main areas in a container terminal.

performed by gantry cranes or by straddle carriers. Stacking cranes could be rail-mounted gantry cranes or rubber-tired gantry cranes. The stacking area is sometimes also called the yard area.

- Landside area

  The landside area is connected to the mainland where trucks pass through gates via roads and trains are both loaded and unloaded by gantry cranes.

- Quayside transport area

  The quayside transport area connects the quayside and the stacking area, involving a number of vehicles for transporting containers. The vehicles can be non-lifting vehicles (e.g., AGVs or trucks) or lifting vehicles (e.g., ALVs or straddle carriers). A group of vehicles owned by the terminal is referred to as the vehicle fleet.

- Landside transport area

  Between the yard area and the landside area, containers are moved by trucks with trailers, multi-trailers, or straddle carriers.

The operations of unloading and loading containers at a container terminal can be described as follows. When a vessel arrives at a berth, the containers have to be taken off the vessel by quay cranes. Then each container is transported by a vehicle from the quayside

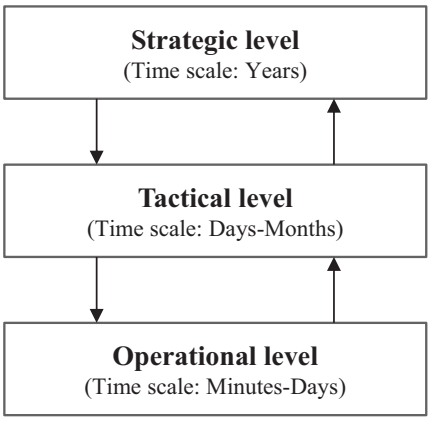


Figure 2.3: The decision problem of container terminals at different levels.

area to the yard area after being unloaded from a quay crane. A stacking crane will pick up the container and move it to a position in the stacking area. Later the containers are retrieved by a stacking crane from the stacking area and transported to another mode, such as barge, train or truck.

A container terminal can be categorized as a manual container terminal, an automated container terminal or a semi-automated container terminal. In a manual container terminal the handling machines of containers are operated by humans. In an automated container terminal all pieces of equipment used for transporting containers are automated, which minimizes the use of human operations. A semi-automated container contains a combination of manual operations and automated operations.

The performance of container terminals can be evaluated using various key performance indicators (KPIs). A primary performance indicator is the turnaround time of the vessel [53, 93, 94], as emphasized by both shippers or terminal operators. The turnaround time is related to other performance indicators that link the transport processes of the terminal directly, e.g., completion time [14, 17, 18], energy consumption [84, 99], vehicle driving distance [31], etc.

2.2 Review on container terminal handling

This section reviews state-of-the-art technologies for container terminal handling in the literature. The first part of this section discusses the decision problems at three different levels and the second part focuses on the operational control of container terminals.

2.2.1 Decision problems

Decision problems at container terminals can be categorized into three levels: strategic, tactical and operational [47, 80, 101] according to the time horizon involved, as shown in Fig. 2.3.

Decision problems at the strategic level concern the layout of the terminal and equipment selection of the terminal, which can be used for a couple of years. For instance, Zhen et al. [121] compared two types of automated container terminals and evaluated the performance of these two types quantitatively; Vis et al. [102] studied two types of automated vehicles (AGVs and ALVs), performing a feasibility and economic analysis on these two types of vehicles.

Tactical problems typically focus on the capacity level of equipment and determine the necessary number of the piece of equipment for completing operations efficiently, ranging from days to months. In the recent literature Alessandri et al. [2] proposed a dynamical approach for determining the percentage of available resources for a particular carrier of one modality using the discrete-time flow model. The number of AGVs in a semi-automated container terminal is determined using a minimal flow algorithm in [103].

At the operational level, the detailed operation of equipment for transporting containers should be decided, in which the timescale varies from minutes to days. The decisions include which piece of equipment processes which container and which route is chosen for transporting containers. The operational decision problem involves the most complex process of the terminal operation and has received significantly attention in the transportation society [93, 94]. In the following part, we review the various approaches for the operational decision problem closely related to the scope of this thesis.

2.2.2 Approaches for operational control

At the operational level, approaches for solving decision problems of container terminals can be categorized as programming-based approaches and analytical approaches. In the following part, we will review these two types of approaches separately.

Programming-based approaches

The programming-based approaches use a computer language to describe the behavior of equipment for handling containers in a container terminal. In the literature, the object-oriented language and the agent-oriented language are the mainstream concepts in the category of programming-based approaches.

- Object-oriented programming

  Object-oriented approaches provide a programming paradigm using the concept of "object" for modeling the terminal. In the object-oriented approach, an object is an entity contains a set of attributes and a set of methods. Attributes are factual descriptions of the object and the methods are functions that enable the object to manipulate its attributes and communicate with other objects [71]. In the object-oriented approach, all physical and conceptual entities can be considered objects [116]. When it comes to a container terminal, each component of the terminal (e.g., one piece of equipment or a vessel) can be modeled as an object. Based on the objects that describe equipment, a container terminal is constructed.

  Regarding the object-oriented programming approach, Duinkerken et al. [29] develop a simulation model for a large-scale automated container terminal based on a traffic control engineering system TRACES that guarantees safety routing; this model is

used for validating experiments and sensitivity analysis of parameters. Bielli et al. [7] develop a distributed discrete-event simulation model using the object-oriented programming for evaluating different operation policies and resource allocation procedures of the container terminal. In [45] a detailed container terminal model based on an object-oriented simulation model Plant Simulation is presented and the performance of container terminals are analysed by varying the speed of equipment.

The object-oriented approaches focus on developing a decision support system for simulating container terminals, in which the effect of different operation policies and parameters on the performance of the terminal can be evaluated. The detailed control and optimization algorithm can be incorporated as the operation policy of the decision support system.

- Agent-oriented programming

  Agent-oriented programming focuses the concept of "agent" and the cooperation of multiple agents, typically referred to as a multi-agent system. In the agent-oriented programming, an agent is a computer system that is capable of independent action on behalf of its user or owner and a multi-agent system consists of a number of agents which interact with each other, typically by exchanging messages [68]. Similarly to the object-oriented programming, each component of a container terminal can be represented by an agent. The agent-oriented approach simulates the simultaneous operations and interactions of multiple agents, in which each agent strives to complete their specified goals by communicating, coordinating and negotiating with other agents and eventually improve the performance of the terminal.

  Several works have been carried out with respect to the agent-based programming. Henesey [48] proposes a multi-agent approach that aims at improving the performance of the container terminal from the terminal manager's perspective by means of increasing the capacity of available resources; however, the focus is on the multi-agent architecture among the pieces of equipment of the container terminal, rather than the control and optimization algorithm of the equipment. Later, Henesey et al. [49] use an agent-based simulator and evaluated operational policies in transshipping containers with real data for verification. Xiao et al. [106] propose a distributed agent system for port planning and scheduling of the berth allocation and requirements for shuttles, in which a large complex problem can be decomposed into a few smaller and manageable ones with information exchange between the agents, resulting in more efficient management. The coordination and cooperation are addressed for berth allocation without considering the detailed coordination between individual pieces of equipment at the operational level.

  The agent-oriented approaches concentrate on investigating the architecture of the multi-agent system for a container terminal, resulting in an intelligent decision support system by means of communication, coordination and cooperation between intelligent agents. Nevertheless, the optimal coordination between different pieces of equipment at the operational level is not addressed using the agent-oriental approach.

Analytical approach

Besides programming-based approaches, analytical approaches model and optimize the operations mathematically, in which equipment scheduling and vehicle management are typically considered separately. Equipment scheduling is closely related to turnaround time of the vessel and it therefore determines operations of equipment at particular times for completion of all containers. Vehicle management focuses the assignment and routing of vehicles. Although it is somehow overlapped with equipment scheduling, vehicle management is discussed in another division in particular due to its high flexibility and complexity.

- Equipment scheduling

  At container terminals, the turnaround time of the vessel is a primary performance indicator for terminal operators [9, 23]. Therefore, the pieces of equipment of the container terminal need to be employed optimally by minimizing the completion time of handling containers. This motivates the investigation of the scheduling problem in which a number of jobs (e.g., containers) are assigned to available resources at particular times for the minimization of the turnaround time.

  Due to the complexity of the container terminal operation, scheduling problems of a particular area are investigated for simplifying the scheduling of the overall terminal. For the quayside area, the quay crane scheduling problem determines the sequence of the QCs' handling jobs and time points at which these are performed [8, 26, 60, 64, 119], considering different objectives and various operations constraints. Although the landside is not directly related to the quayside operation, the yard crane scheduling problem of a single block [36, 37, 75] and multiple blocks [16, 46] have been studied for improving operation efficiency of the stacking area.

  However, the transport of a container depends on the interaction of multiple machines from areas all over the container terminal. The individual scheduling of equipment may lead to the loss of the overall performance. This motivates the research of integrated scheduling of multiple areas. Cao et al. [14] consider the integrated scheduling of the quayside transport area and the stacking area. Furthermore, the quayside area, the quayside transport area and the stacking area together constitute the transport of containers between the vessel and the stack and therefore the integrated scheduling of these three areas has been investigated [17, 18, 62, 72].

  The scheduling problem of equipment aims at minimizing the completion time of all jobs, which directly or indirectly reduces the turnaround time of the vessel. The scheduling problem has been integrated for optimally coordinating different types of equipment. Still, the scheduling problem is concerned with productivity improvement only without considering energy efficiency in the scheduling problem. Furthermore, the integrated scheduling problem is typically formulated as an open-loop control problem and therefore cannot handle uncertainties in the terminal environment.

- Vehicle management

  In the quayside transport area, a number of vehicles are used for transporting containers between the quayside area and the stacking area. The employed vehicles can be self-lifting vehicles (e.g., straddle carriers and ALVs) and non-lifting vehicles (e.g.,

yard trucks and AGVs) [15]. At the operation level, the vehicle management problem involves determining which vehicle transports which container and which route is chosen [15, 101]. Typically the route of the vehicle is planned when the assignment decision has been made [100].

The first problem, referred to as the assignment problem, is to assign a container to a particular vehicle by maximizing or minimizing a defined objective function. The objective function can be the minimal delay time, operation cost, completion time and etc [100]. As a type of non-lift vehicles, AGVs are widely used in automated container terminals and therefore quite a few works have been done with respect to the assignment of AGVs. In [59] a look-ahead dispatching methodology is suggested by minimizing the total delay of crane quays with a small penalty on the total distance of AGVs. Grunow et al. [43] investigate a real-time dispatching of multi-load AGVs using a sequential coordination scheme for different types of equipment. Briskorn et al. [10] also consider a real-time dispatching method based on an analogy to inventory management. Angeloudis et al. [3] propose an AGV assignment algorithm by maximizing the total benefit taking some uncertainties into account, which is suitable for real-time of AGVs. Besides the non-lift vehicles, the assignment of self-lifting AGVs is also considered. Cai et al. [11] propose rescheduling policies for large-scale task assignment of autonomous straddle carriers under certainties.

It is noted that the vehicle assignment problems of the quayside transport area are considered preferably under uncertainties for adjusting changes in the terminal environment in real-time. The assignment problem of AGVs is mostly considered as an individual research problem using different simplification procedures. However, due to its strong interconnection with the quayside area, the quayside handling capacity should be demonstrated explicitly using a particular algorithm for assigning the AGVs.

The second problem is the routing problem which focuses on avoiding collision and deadlock between different autonomous vehicles (e.g., AGVs). The routing layout involved typically considers the mesh routing and the free-ranging routing. The mesh routing searches the shortest route between an origin and a destination through a fixed path in the mesh, while the free-ranging routing allows a free-travel trajectory instead of sticking to a fixed path. For the mesh routing problem, Kim et. al [58] develop an efficient deadlock prediction and prevention algorithm of AGVs by occupying more grid-blocks for each vehicle; Zeng et al. [118] develop a discrete-time model for general container routing in mesh route layouts and proposed a routing algorithm for collision avoidance by allowing the vehicle to change its velocity at the interactions; Gawrilow et al. [34] suggest a conflict-free dynamic routing algorithm incorporating the time-dependent model of AGVs and therefore both the conflict and deadlock can be prevented when the route is made. When it comes to the free-ranging routing, Duinkerken et al. [30] propose an approach for collision-free trajectory planning of AGVs that aims to complete the operations within the given scheduled windows; however, the quayside handling capacity is not clearly indicated.

The mesh routing problem and the free-ranging routing problem are investigated for preventing collision and deadlocks of vehicles during the transport process in the quayside transport area. The trajectory planning of free-ranging AGVs for a high

quayside handling capacity has hardly been investigated due to the complexity of the integrated planning problem.

Summary

Programming-based approaches and analytical approaches are two main categories for addressing the operational decision problems of container terminals. The programming-based approach aims to develop a decision support system for evaluating and analysing the performance of the terminal, in which the detailed control and optimization algorithm can be incorporated as the operation policies of the decision support system. The analytical approach focuses mathematical optimization for determining the detailed scheduling, assignment and routes of equipment for maximizing the performance of the terminal. The existing scheduling problem are integrated with different types of equipment due to its interaction, but the focus is on only productivity without considering energy efficiency. There is little attention paid on trajectory planning of free-ranging AGVs and further investigation on implementing free-ranging AGVs for the high quayside handling capacity is needed.

2.3 Control for large-scale systems

A container terminal is in this review regarded as a large-scale system, in which each piece of equipment has its own dynamics and different pieces of equipment coordinate with each other. To cope with new challenges of container terminals, advanced methodologies for controlling these pieces of equipment have to be developed. This section reviews advanced control approaches for large-scale systems, detailing the link between advanced control approaches for large-scale systems and the container terminal further.

A large-scale system includes a large number of subsystems and each subsystem interacts with other subsystems. The control of such a large-scale system bring great challenges to control engineers due to its computation complexity and communication limits [67, 92]. For addressing these challenges, distributed control and hierarchical control [67, 87, 92] decompose the complexity for controlling the large-scale system and these advanced has been applied into various application areas, like power networks [4, 33, 55], water networks [27, 74, 76, 77] and transportation networks [24, 73, 79], etc.

2.3.1 Centralized control

A centralized control system considers an unique controller for the whole system. The decisions of the centralized controller are determined based on a model that describes the dynamics of the system as a whole assuming all the information of the system is available. The centralized control structure is given in Fig. 2.4.

Centralized control is regarded as a classical approach for controlling everything considered in the system naturally. However, control of a large-scale system requires new techniques. As the scale of the system increases, the centralized controller is not realistic due to the amount of information to be considered for modeling and measurements [67]. Instead, the overall system has to be decomposed into several subsystems in which each subsystem is controlled locally via communication. This motivates the development of distributed control and hierarchical control.

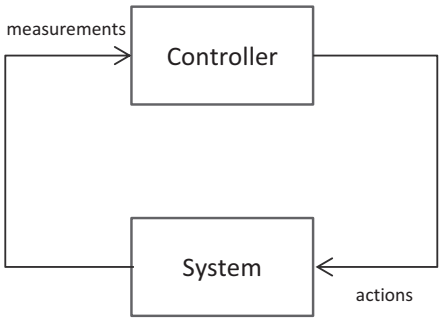


Figure 2.4: The centralized control structure.

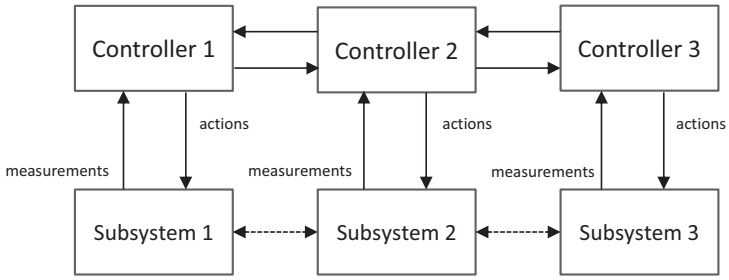


Figure 2.5: A distributed control structure (The dashed line indicates possible interaction).

In the domain of container terminals, most of analytical approaches for solving operational decision problems use the centralized control framework. For instance, the scheduling problem of equipment is determined by a centralized controller.

2.3.2 Distributed control

A distributed control system is a control system, wherein each subsystem is controlled by a controller locally and each local controller coordinates with each other. In distributed control problems the local controllers are completely or almost independent [67] and therefore a distributed control system is typically formulated in a single level. A distributed control structure can be shown in Fig. 2.5,

Distributed model predictive control [13, 69] is regarded a powerful approach for addressing large-scale complex problems. Theoretical developments on distributed model predictive control have grown rapidly recently, including several fundamental issues, e.g., decoupling [56, 104], stability [95] convergence [38], communication [70] and etc. Furthermore, distributed model predictive control has been applied into power systems [4], water systems [27, 74], multiple robots [57, 61], transportation systems [24], etc.

In container terminals, although Van Dam [22] discuss the potential of distributed control for trajectory planning of free-ranging AGVs, there is no particular paper that addresses operational problems using the distributed control.

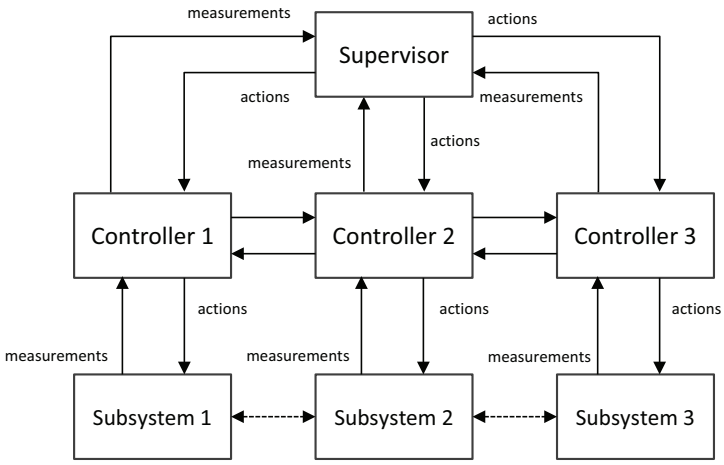


Figure 2.6: A hierarchical control structure. (The dashed line indicates possible interaction).

2.3.3 Hierarchical control

Hierarchical control addresses the situation in which local controllers are not independent but have to respond to the data of other local controllers [67], which is different from the distributed control system. Considering this property, a large-scale system is decomposed into several levels using a hierarchical structure for the coordination of difference levels. Therein a large control problem is decomposed several smaller sub-problems for solving. A typical hierarchical control structure is illustrated in Fig. 2.6.

Hierarchical control has been applied into many applications. In the domain of power systems, Kamwa et al. [55] develop a hierarchical approach for stabilizing control of large power systems; Edlund el al. [33] investigate a hierarchical model-based predictive control for predicting and control of the renewable energy generation; in [44] a hierarchical control architecture for intelligent microgrids is proposed for integrating distributed renewable energy sources. When it comes to water management, Zafra-Cabeza et al. [117] apply the hierarchical model predictive control into the irrigation canal planning; Ocampo et al. [76, 77] investigate the multi-level decentralised model predictive control for drinking water networks. In transportation applications, Papamichail et al. [79] propose a model-predictive hierarchical control approach for coordinating ramp metering of freeway networks; Zhang et al. [120] develop a hierarchical decentralized decision architecture for path planing of a large number of flights; Nabais et al. [73] suggest a hierarchical model predictive control using a flow perspective for modality exchanges of the freight transport network.

In container terminals, the programming-based approaches, either using the object-oriented language or the agent-oriented language, considers a hierarchical control structure. In the object-oriented programming objects are defined in a hierarchical environment [116]. The agent-oriented programming does not specialize the hierarchy in the definition of agents. However, the existing agent-based programming approaches with respect to container terminals typically propose the hierarchical architecture, e.g., [49, 106].

2.3.4 Summary

Three control structures for addressing the large-scale systems are discussed. Due to the overwhelming amount information for modelling and measurements, distributed control and hierarchical control are thereby considered for decomposing the complexity. In the existing research on container terminals, the programming-based approaches use the hierarchical architecture while the analytical approaches mainly address the control problem in a centralized way.

2.4 Benchmark systems

Benchmarks are used for measuring the performance of container terminals and identifying the best methodology in order to improve the performance and increase productivity [23, 91]. Thus, benchmarking the terminal operation is indispensable and terminal operators can judge whether a proper performance is achieved at their terminal. In literature, the available benchmark for container terminals focuses on strategic financial performances using data envelopment analysis [23, 91], while this thesis focuses on detailed technical operation. Therefore, regarding the performance of the terminal operation, three benchmarks with respect to three terminal layouts are proposed in this section. The proposed benchmark systems are used for evaluating the performance of the container terminal when different control methods are considered.

We consider three types of container terminals: a compact terminal, a medium terminal and a large terminal. The compact terminal is the most basic case, including the typical component of a QC, an AGV and an ASC. As the extension of the compact case, the medium-sized terminal involves a QC, multiple AGVs and multiple ASCs. Furthermore the large-size terminal contains multiple QCs, multiple AGVs and multiple ASCs. These three different terminal layouts for each benchmark system are illustrated in Fig. 2.7, Fig. 2.8 and Fig. 2.9, respectively.

2.4.1 Features

The features of the benchmarks are based on a typical container terminal layout provided by a consultancy company [96] and information sheets of equipment [41, 42, 54]:

- The stowage width of the container vessel is assumed to be 8 TEU;
- The vessel is assumed to be in the berth ready for loading and unloading;
- The distance between the furtherest container and the interchange point of the QC is 100 meters;
- The quayside transport area is 150 m$\times$270 m;
- Each stack has the length of 36 TEU, the width of 10 TEU and the height of 6 TEU for capacity;
- The maximum speed of the QC, AGV and ASC are assumed to be $v_{\max}^{\mathrm{qc}} = 4$ [m/s], $v_{\max}^{\mathrm{agv}} = 6$ [m/s] and $v_{\max}^{\mathrm{asc}} = 4$ [m/s], respectively;

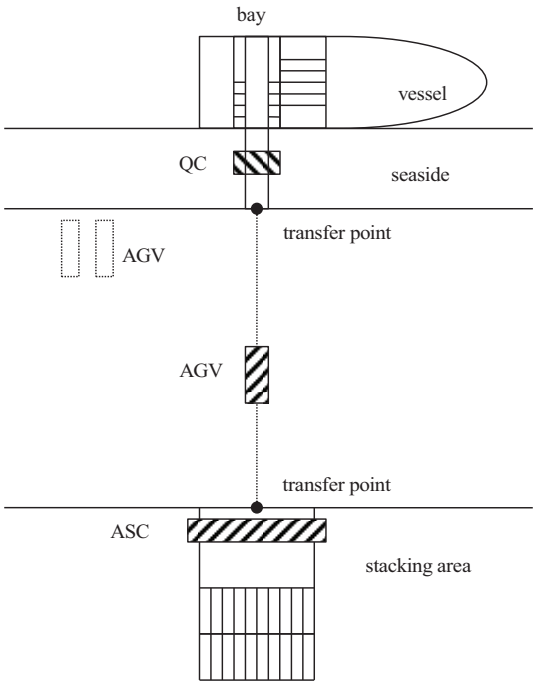


Figure 2.7: The layout of Benchmark System 1.

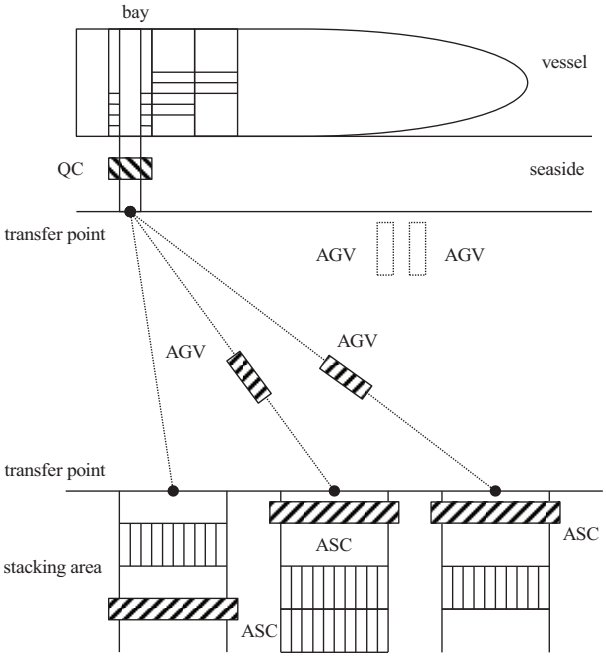


Figure 2.8: The layout of Benchmark System 2.

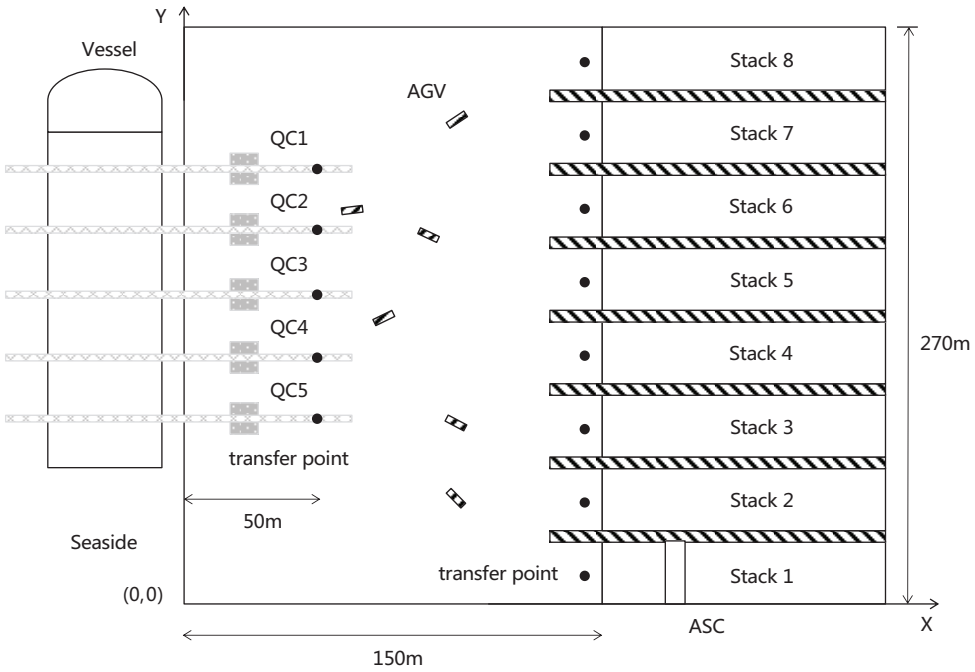


Figure 2.9: The layout of Benchmark System 3.

- The maximum acceleration of the QC, AGV and ASC are assumed to be $u_{\max}^{\mathrm{qc}} = 0.4$ [m/s^2], $u_{\max}^{\mathrm{agv}} = 1$ [m/s^2] and $u_{\max}^{\mathrm{asc}} = 0.4$ [m/s^2], respectively;
- Each piece of equipment can only transport one container at a time;

2.4.2 Key performance indicators

In container terminals, the performance of container terminals can be evaluated using various indicators [80]. Thereby for answering the key research questions of this thesis, it is necessary to define key performance indicators (KPIs) that measures the operation efficiency and evaluate the performance of the proposed control algorithm. First of all, a primary performance indicator is the turnaround time of the vessel [53, 93, 94], as emphasized both for shippers or terminal operators. The turnaround time is related to other performance indicators that link the transport processes of the terminal directly, e.g., completion time [14, 17, 18], energy consumption [84, 99], vehicle driving distance [31], etc. The detailed KPIs are defined as follows:

- KPI 1: Turnaround time [h]

  Turnaround time is defined as the time a vessel spends at the berth for the purposed of loading and unloading [53]. The turnaround time is well-recognized as an important factor in the overall transport cost of containers. It is a primary performance indicator for terminal operators.

- KPI 2: Completion time [s]

  Completion time is the ending time for loading and unloading containers of the vessel using the pieces of equipment at the terminal, which is directly related to the turnaround time of the vessel.

- KPI 3: Energy consumption [kWh]

  Energy consumption refers to energy of the employed pieces of equipment used for transporting containers.

- KPI 4: Computation time [s]

  Computation time is the time spent for solving a particular optimization problem with regard to container handling.

- KPI 5: AGV average traveling distance [m]

  This distance is the average distance of the AGV which moves between the transfer point at the quayside and the transfer point at the stack for transporting containers.

- KPI 6: AGV relative distance [m]

  The relative distance of AGVs is considered the distance between 2 AGVs that are used for transporting containers.

- KPI 7: QC utilization [%]

  QC utilization refers to the percentage of time during which QCs are active on average.

- KPI 8: AGV utilization [%]

  AGV utilization refers to the percentage of time during which AGVs are active on average.

- KPI 9: ASC utilization [%]

  ASC utilization refers to the percentage of time during which ASCs are active on average.

KPI 1-3 and KPI 7-9 are related to key research question 1, KPI 4 is linked to key research question 2 and KPI 1, 2, 5, 6 are associated with key research question 3.

2.5 Summary

In this chapter an overview of container terminal operations is presented. The research on decision problems of container terminals and advanced control technologies for large-scale systems have been reviewed. Three benchmarks with respect to different terminal layouts have been proposed at the end of the chapter.

In Chapter 3, 4, 5 and 6 novel approaches for controlling the proposed benchmark systems will be presented. Chapter 3 is linked to Benchmark System 1 for the compact container terminal. Chapter 4 and 5 are related to Benchmark System 2 with respect to the medium container terminal. Chapter 6 is concerned with Benchmark System 2 for large container terminal. These benchmark systems will be assessed using the listed KPIs.

Chapter 3

Hybrid MPC for energy efficiency

As has been discussed in Chapter 1, energy efficiency has become a practical problem for terminal operators. Energy consumption is expected to reduced while still achieving high handling capacity when a number of containers are transported in the container terminal. Chapter 2 indicates that energy efficiency of container terminals has been addressed merely at the strategic level, instead of at the operational level. This chapter is going to investigate an approach for improving energy efficiency of the compact container terminal at the operational level.

The research discussed in the chapter is based on [107, 108, 115].

3.1 Introduction

In the quayside operation of automated container terminals, QCs, AGVs and ASCs are operated cooperatively for loading or unloading a vessel. As the most basic configuration of interactive operations, the compact container terminal considers the case of the one QC, one AGV and one ASC. The investigation of improving energy efficiency for the compact terminal is valuable when it comes to a general container terminal. The exiting literature with respect to the operational control of container terminals mainly focuses on the discrete-event dynamics of the pieces of equipment when containers are transported (e.g., [14, 29]). For the compact terminal, the control of the piece of equipment is hereby addressed in a distributed way. The discrete-event dynamics drives the two interconnected piece of equipment for loading or unloading a container, while the control of continuous-time dynamics is simplified as a fixed driving behavior locally [14, 29].

The control structure of this distributed controller is presented in Fig. 3.1. The interaction of different pieces of equipment follows the discrete-event dynamics and the controller of each piece of equipment coordinates for loading and unloading containers. The continuous-time dynamics of the piece of equipment are controlled locally.

Energy efficiency is concerted with both the handling capacity and the energy consumption. The handling capacity depends on the discrete-event dynamics while energy consumption is determined by the continuous-time dynamics in which the position and the speed change over time. Discrete-event dynamics and continuous-time dynamics must be

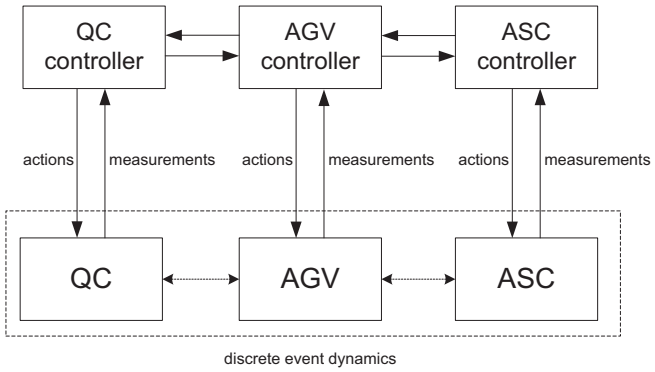


Figure 3.1: A distributed control structure for the compact terminal.

considered together for improving energy efficiency of the operational control at container terminals.

At the operational level, energy efficiency is expected to be obtained for real-time operation. In real-time operation uncertainties (e.g., operation delays and the precise time at which new containers arrive) can change the process of transporting containers and influence energy efficiency of the container handling system. To adjust changes in the dynamically operating environments of container terminals, real-time decisions for energy efficiency need to be determined.

This chapter proposes a methodology for improving energy efficiency of the compact terminal during real-time operation, in which uncertainties can be handled directly. For energy efficiency, the combination of discrete-event dynamics and continuous-time dynamics, referred to as hybrid systems, is modeled using interconnected hybrid automata. After transforming the hybrid automata into logical dynamical models, a hybrid Model Predictive Control (MPC) controller is proposed for real-time operation. The underlying control problem is hereby formulated as a mixed integer linear programming problem that can be solved by efficient solvers.

The remainder of this chapter is organized as follows. Section 3.2 presents the modeling formalism using hybrid automata and its transformation into mixed logical dynamical models. Section 3.3 subsequently proposes a hybrid model predictive controller for achieving energy efficiency. Section 3.4 discusses the results of the simulation experiments and demonstrates the performance of the proposed hybrid MPC controller when facing two types of uncertainties. Section 3.5 concludes this chapter.

3.2 Hybrid automaton modeling formalism

Since the system under study involves a combination of discrete-event dynamics and continuous-time dynamics, we propose to represent the dynamics using a hybrid automaton [50, 89]. A general hybrid automaton H can be defined as $H = (S, X, U, f, \text{Init}, \text{Inv}, E, G, R)$, where

- S is a finite set of discrete modes;
- X is a finite set of continuous state variables;

- U is a finite set of control variables;
- $f : S \times X \times U$ describes the evolution of continuous variables in a certain discrete mode. The evolution of the continuous state depends on the discrete mode and the action.
- Init is the set of possible initial states;
- $\text{Inv} : S \to P(X)$ describes the invariant set that defines the feasible regions of continuous variables in a certain discrete mode, where $P(X)$ denotes the power set (set of all subsets) of X;
- $E : S \times S$ is the set of edges representing the possible switches between discrete modes;
- $G = G(s^{\alpha}, s^{\beta}) : S \to P(X,U)$ serves as the guard giving conditions for when the discrete mode transitions from s^{α} to s^{β};
- $R : E \times X \to P(X)$ resets the continuous variables between the switches of discrete modes.

3.2.1 Interconnected hybrid automaton

In our case, we consider sets of inter-connected hybrid automata. The automata interact via the guards: transitions between certain discrete modes are only possible when guards involving variables from multiple automata are satisfied. For this, we need to extend the description of the general hybrid automaton. The hybrid interconnected automaton is described by:

$H^{\text{inter}} = (S, X, U, f, \text{Init}, \text{Inv}, E, G, R, V, G^{\text{inter}})$, where besides the components of a general automation,

- V is a finite set of variables of other hybrid automata;
- $G^{\text{inter}} = G^{\text{inter}}(s^{\alpha}, s^{\beta}) : S \to P(X,U)$ is an interconnecting guard, i.e., a guard involving variables from X and U as well as V.

For an interconnected hybrid automaton, the discrete mode in S and the state of variables in X for itself and the state of variables in V from other hybrid automata can trigger an interconnecting guard G^{inter}. G^{inter} indicates the guard in which the other interacting machine is involved. After G^{inter} is triggered, the discrete mode can switch from one to another. By introducing V and G^{inter}, the interaction between two interacting machines can be presented more clearly. For example, an interacting guard can represent the moment at which it becomes possible for an AGV to transfer one container to an ASC. More details will be discussed later.

3.2.2 Modeling of components

We make a distinction between controlled and uncontrolled components when a container is transported from a bay of the vessel to a stack in the stacking area. A QC, an AGV and an ASC are referred to as controlled components because actions of these pieces of equipment need to be determined by a controller. The vessel and the stack are regarded as uncontrolled components since the vessel and stack do not move during the time of transport.

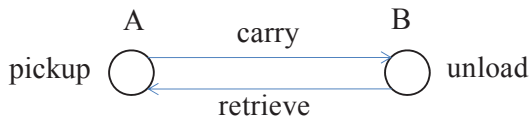


Figure 3.2: The general model of a controlled component.

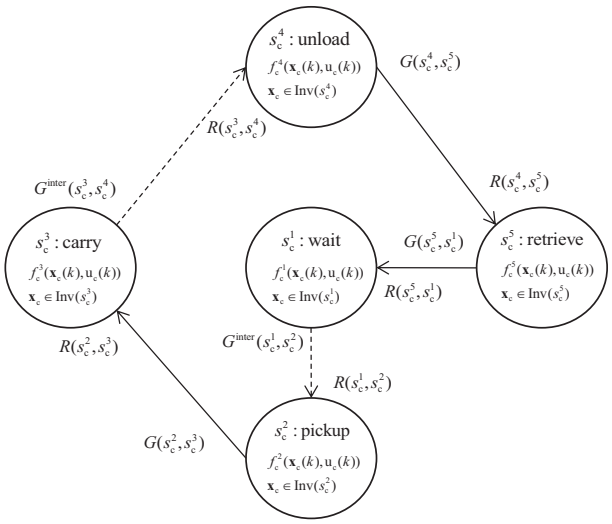


Figure 3.3: The hybrid automaton of a controlled component. The dashed line indicates that the guard depends on the availability of another component.

Modeling of controlled components

The QC, AGV and ASC can be modeled in a generic way as controlled components that transport a container between two points: one where a component picks up or receive the container and one where it unloads or offers the container, as illustrated in Fig. 3.2.

The controlled component picks up one container at position A and transports the container from A to B, where it is unloaded. The dynamics of one controlled component can be described as an interconnected hybrid automaton shown in Fig. 3.3. The details of the hybrid automaton are presented as follows:

$H_c = (S_c, X_c, U_c, f_c, \mathrm{Init}_c, \mathrm{Inv}_c, E_c, G_c, R_c, V_c, G_c^{\mathrm{inter}})$,
where

- $S_c = \{s_c^1, s_c^2, s_c^3, s_c^4, s_c^5\}$ gives five discrete modes in which one controlled component can be. In the mode s_c^1 (wait), this component waits for one interacting component to pick up a container (s_c^2). In the mode s_c^2 (pickup), this component picks up a container at place A. In the mode s_c^3 (carry), this component is moving the container from A to B. In the mode s_c^4 (unload), this component unloads the container at place B when another interacting component is available to unload the container. In the mode s_c^5 (retrieve), this component is moving from B to A to pick up a container at place A after unloading a container at place B.

- $X_c = \{x_c^{pos}(k), x_c^{vel}(k)\}$ $(x_c^{pos}(k) \in \mathbb{R}, x_c^{vel}(k) \in \mathbb{R})$ is the set of continuous states: the position $x_c^{pos}(k)$ (m) and the velocity $x_c^{vel}(k)$ (m/s) of the component.

- $U_c = \{u_c(k)\}$ (m/s^2) is the set of control variables, representing the acceleration of the component.

- f_c describes the continuous-time dynamics in each discrete mode. We define Δt as the sampling time. Let $\mathbf{x}_c(k) = \begin{bmatrix} x_c^{pos}(k) & x_c^{vel}(k) \end{bmatrix}^T$. Then the continuous dynamics per mode are modeled as follows:

  – In mode 1 (wait), mode 2 (pickup) and mode 4 (unload): the position and the speed of the component do not change. Therefore, the discretized continuous-time dynamics with respect to these three modes $f_c^1(\mathbf{x}_c(k), u_c(k))$, $f_c^2(\mathbf{x}_c(k), u_c(k))$ and $f_c^4(\mathbf{x}_c(k), u_c(k))$ are described as:

  $$\mathbf{x}_c(k+1) = \mathbf{x}_c(k). \tag{3.1}$$

  – In mode 3 (carry) and mode 5 (retrieve): we consider the double integrator as the continuous-time dynamics, without consideration for air-drag and rolling resistance. Therefore, the discretized continuous-time dynamics in mode 3 and mode 5, namely $f_c^3(\mathbf{x}_c(k), u_c(k))$ and $f_c^5(\mathbf{x}_c(k), u_c(k))$, are presented as:

  $$\mathbf{x}_c(k+1) = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \mathbf{x}_c(k) + \begin{bmatrix} 0.5\Delta T^2 \\ \Delta T \end{bmatrix} u_c(k). \tag{3.2}$$

- Inv_c is defined for this controlled component as follows:
  $\text{Inv}(s_c^1) = \{x_c^{pos}(k) = x_c^{unload}\}$,
  $\text{Inv}(s_c^2) = \{x_c^{pos}(k) = x_c^{unload}\}$,
  $\text{Inv}(s_c^3) = \{x_c^{load} \le x_c^{pos}(k) \le x_c^{unload}\}$,
  $\text{Inv}(s_c^4) = \{x_c^{pos}(k) = x_c^{load}\}$,
  $\text{Inv}(s_c^5) = \{x_c^{load} \le x_c^{pos}(k) \le x_c^{unload}\}$,
  where x_c^{load} and x_c^{unload} are positions for loading and unloading containers.

- E_c is defined as the set:
  $\{(s_c^1, s_c^2), (s_c^2, s_c^3), (s_c^3, s_c^4), (s_c^4, s_c^5), (s_c^5, s_c^1)\}$

- G_c has the following guards with respect to this controlled component: $G(s_c^1, s_c^2) = \{s_c(k) = s_c^1, x_c(k) = x_c^{load}\}$. This guard depends on the availability of another component for picking up a container. This dependence is represented by the dashed line in Fig. 3.3. $s_c(k)$ is referred to as the discrete mode of the component at time k
  $G(s_c^2, s_c^3) = \{s_c(k) = s_c^2\}$, i.e., the component finishes the pickup.
  $G(s_c^3, s_c^4) = \{s_c(k) = s_c^3, x_c(k) = x_c^{unload}\}$, i.e., this component reaches the loading position and wait for being unloaded. This guard also depends on the availability of another component for unloading a container, as presented in Fig. 3.3.
  $G(s_c^4, s_c^5) = \{s_c(k) = s_c^4\}$, i.e., the component finishes unloading.
  $G(s_c^5, s_c^1) = \{x_c(k) = x_c^{load}\}$, i.e., this component reaches the loading position.

Table 3.1: G_c^{inter} and its coupled interconnected guards.

G^{inter}	coupled G^{inter}
$G^{inter}(s_{qc}^1, s_{qc}^2)$	$G^{inter}(s_{agv}^3, s_{agv}^4)$
$G^{inter}(s_{agv}^1, s_{agv}^2)$	$G^{inter}(s_{asc}^3, s_{asc}^4)$

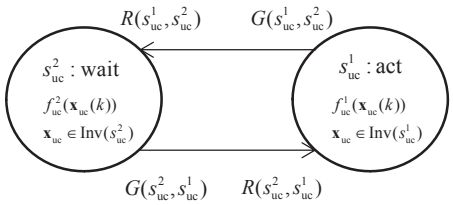


Figure 3.4: The hybrid automaton for an uncontrolled component.

- The continuous state does not change as a result of switching the discrete modes. Therefore,

  $R_c = \left\{(x_c^-, x_c^+) \mid x_c^- \in \mathbb{R}^2, x_c^+ \in \mathbb{R}^2 \text{ and } x_c^- = x_c^+\right\}$.

- V_c is associated with variables of other hybrid automata interacting with this interconnected hybrid automaton. The states of interacting variables of other hybrid automata are used to trigger the interconnecting guards.

- G_c^{inter} describes the guard of controlled components interacting with different hybrid systems simultaneously. This indicates two G_c^{inter} of each interconnected hybrid automaton are coupled. Here, the controlled components are QC, AGV and ASC. Therefore, $G^{inter}(s_c^\alpha, s_c^\beta)$ is extended as $G^{inter}(s_{qc}^\alpha, s_{qc}^\beta)$, $G^{inter}(s_{agv}^\alpha, s_{agv}^\beta)$ and $G^{inter}(s_{asc}^\alpha, s_{asc}^\beta)$. Specifically, the container is transferred from the QC to the AGV in which $G^{inter}(s_{qc}^1, s_{qc}^2)$ and $G^{inter}(s_{agv}^3, s_{agv}^4)$ are triggered at the same time. Similarly, $G^{inter}(s_{agv}^1, s_{agv}^2)$ and $G^{inter}(s_{asc}^3, s_{asc}^4)$ are triggered simultaneous when a container is transported from the AGV to the ASC. The G_c^{inter} and its coupled interconnected guards of controlled components is shown in Table 3.1. Besides this, the guard G_c^{inter} and its coupled interconnected guards of uncontrolled components will be discussed later.

Modeling of uncontrolled components

The dynamics of an uncontrolled component can be described as a hybrid automaton in the following way (see Fig. 3.4):

$H_{uc} = (S_{uc}, X_{uc}, f_{uc}, \mathrm{Init}_{uc}, \mathrm{Inv}_{uc}, E_{uc}, G_{uc}, R_{uc})$, where

- S_{uc}=$\left\{s_{uc}^1, s_{uc}^2\right\}$ gives two discrete modes in which an uncontrolled component can be. In the discrete mode s_{uc}^1 (act), one container is loaded or unloaded from this uncontrolled component. In the discrete mode s_{uc}^2 (wait), this component waits until the container is handled.

- $X_{uc} = \{N_{uc}(k)\}$ $(N_{uc}(k) \in \mathbb{R})$

$N_{\mathrm{uc}}(k)$ is the constraints of the number with respect to containers in this component.

- f_{uc} represents the dynamics of this uncontrolled component. Let $x_{\mathrm{uc}}(k) = N_{\mathrm{uc}}(k)$. The continuous dynamics of the two discrete modes are modeled as follows:

  1. In mode 1 (act), the number of containers changes. So $f^1_{\mathrm{uc}}(N_{\mathrm{uc}}(k))$:

  $$x_{\mathrm{uc}}(k+1) = x_{\mathrm{uc}}(k) + a_{\mathrm{uc}}. \tag{3.3}$$

  where in the vessel $a_{\mathrm{uc}} = -1$ and in the stack $a_{\mathrm{uc}} = 1$

  2. In mode 2, the number of containers does not change. So $f^2_{\mathrm{uc}}(N_{\mathrm{uc}}(k))$:

  $$x_{\mathrm{uc}}(k+1) = x_{\mathrm{uc}}(k). \tag{3.4}$$

- $\mathrm{Inv}_{\mathrm{uc}}$ is defined for this uncontrolled component as:
  $\mathrm{Inv}(s^1_{\mathrm{uc}})=\{0 \le x_{\mathrm{uc}}(k) \le N\}$,
  $\mathrm{Inv}(s^2_{\mathrm{uc}})=\{0 \le x_{\mathrm{uc}}(k) \le N\}$,
  where N is the capacity of this component.

- E_{uc} is defined as the set: $E = \left\{(s^1_{\mathrm{uc}}, s^2_{\mathrm{uc}}), (s^2_{\mathrm{uc}}, s^1_{\mathrm{uc}})\right\}$.

- G_{uc} has the following guards with respect to this uncontrolled piece of equipment:
  $G(s^1_{\mathrm{uc}}, s^2_{\mathrm{uc}}) = \left\{x^{\mathrm{pos}}_{\mathrm{uc}}(k) = x^{\mathrm{act}}_{\mathrm{c}}\right\}$. This guard depends on the arrival of the controlled component.
  $G(s^2_{\mathrm{uc}}, s^1_{\mathrm{uc}}) = \left\{s_{\mathrm{uc}}(k) = s^1_{\mathrm{uc}}\right\}$, i.e., the handling of a container finishes. $s_{\mathrm{uc}}(k)$ is used to describe the discrete mode state of the uncontrolled component at time k.

- The continuous state does not change as a result of switching the discrete modes. So $R(s^1_{\mathrm{uc}}, s^2_{\mathrm{uc}}) = R(s^2_{\mathrm{uc}}, s^1_{\mathrm{uc}}) = \{\mathrm{N}^-_{\mathrm{uc}} = \mathrm{N}^+_{\mathrm{uc}}\}$.

- V_{uc} is associated with variables of other hybrid automata that have interaction with this interconnected hybrid automaton.

- $G^{\mathrm{inter}}_{\mathrm{uc}}$ describes the guard of uncontrolled components interacting with controlled interconnected hybrid automata simultaneously. In this chapter, the vessel and the stack are referred to as the uncontrolled components. The term uc can be replaced by v and s to refer to the vessel and the stack. As detailed, $G^{\mathrm{inter}}(s^2_{\mathrm{v}}, s^1_{\mathrm{v}})$ and $G^{\mathrm{inter}}(s^3_{\mathrm{qc}}, s^4_{\mathrm{qc}})$ are coupled when the QC picks up a container from the vessel. Also, $G^{\mathrm{inter}}(s^2_{\mathrm{v}}, s^1_{\mathrm{v}})$ and $G^{\mathrm{inter}}(s^3_{\mathrm{qc}}, s^4_{\mathrm{qc}})$ are synchronized when the ASC unloads a container in the stack. These coupled interconnected guards are shown in Table 3.2.

Table 3.2: $G^{\mathrm{inter}}_{\mathrm{uc}}$ and its coupled interconnected guards

G^{inter}	coupled G^{inter}
$G^{\mathrm{inter}}(s^2_{\mathrm{v}}, s^1_{\mathrm{v}})$	$G^{\mathrm{inter}}(s^3_{\mathrm{qc}}, s^4_{\mathrm{qc}})$
$G^{\mathrm{inter}}(s^2_{\mathrm{s}}, s^1_{\mathrm{s}})$	$G^{\mathrm{inter}}(s^1_{\mathrm{asc}}, s^2_{\mathrm{asc}})$

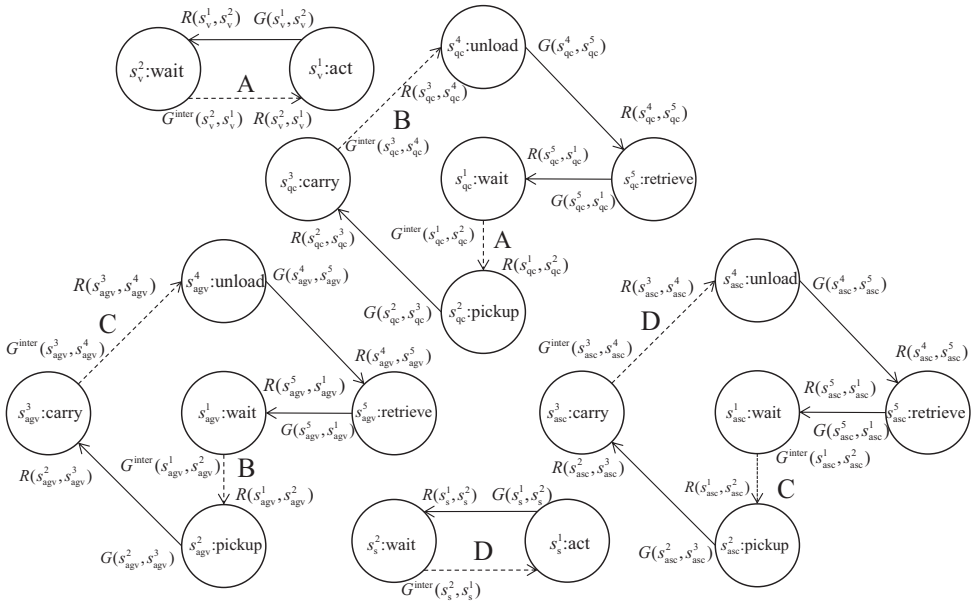


Figure 3.5: Simplified representation of the complete system (continuous-time dynamics are left out for clarity). Five components are coupled by interconnected guards indicated by A, B, C and D.

3.2.3 Modeling of five components combined

This section discusses the transformation of interconnected hybrid automata into mixed logical dynamical models. Based on the interconnected hybrid automata for controlled and uncontrolled components introduced in Section 3.2.2, the compact terminal system is represented as a whole. Then the complete mathematical model of the complete system is described in HYSDEL for generating a mixed logical dynamical model for control purposes.

Interaction of components

Five components are modeled in this chapter. Since we are interested in the controlled components, the time-dynamics of three controlled components (i.e., QC, AGV and ASC) are considered here. The generic abbreviation "c" refers to a controlled component as introduced in Section 2.3.1. The "c" will be replaced by "qc", "agv" and "asc" to specify the explicit controlled components (i.e., QC, AGV and ASC) respectively. Similarly, the abbreviation "v" and "s" replace "uc" to denote the vessel and the stack regarded as the uncontrolled components. When a container is transported from a bay of the vessel to the storage position in a stack, the QC and the AGV are interacting in the quayside area, whereas the AGV is interacting with the ASC in the stacking area. This interaction is shown in Fig. 3.5.

The interactions between two different components are denoted by A, B, C and D in Fig. 3.5, indicating two guards of two interacting components take place simultaneously. For interaction A, the vessel switches from the discrete mode s_v^2 to s_v^1 to dispatch a container

when $G^{\text{inter}}(s_{\text{v}}^2, s_{\text{v}}^1)$ is triggered. At the same time, $G^{\text{inter}}(s_{\text{v}}^2, s_{\text{v}}^1)$ and $G^{\text{inter}}(s_{\text{qc}}^1, s_{\text{qc}}^4)$ coincide when the discrete mode of the vessel will change from s_{v}^2 to s_{v}^1 to pick up a container. Similarly, the synchronization of two interacting components can be specified for B, C and D.

Mixed logical dynamical model

The integration of five components mentioned above forms again a hybrid system, including linear continuous-time dynamics and discrete-event dynamics. Such a class of hybrid systems can be described as mixed logical dynamical systems (MLD) [5], in which the continuous-time part is described by linear dynamics and the discrete-event part is modeled as a set of linear constraints on binary variables and continuous variables. This type of model is well-suited for the formulation of model predictive control problems for hybrid systems [12] as discussed in the next section.

The general MLD model is described as follows [5]:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\mathbf{u}(k) + \mathbf{B}_2\delta(k) + \mathbf{B}_3\mathbf{z}(k) \tag{3.5}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}_1\mathbf{u}(k) + \mathbf{D}_2\delta(k) + \mathbf{D}_3\mathbf{z}(k) \tag{3.6}$$

$$\mathbf{E}_2\delta(k) + \mathbf{E}_3\mathbf{z}(k) \leq \mathbf{E}_1\mathbf{u}(k) + \mathbf{E}_4\mathbf{x}(k) + \mathbf{E}_5, \tag{3.7}$$

where $\mathbf{x}(k) = [\mathbf{x}_r^{\text{T}}(k), \mathbf{x}_b^{\text{T}}(k)]^{\text{T}}$ with $\mathbf{x}_r(k) \in \mathbb{R}^{\text{n}}$ is the continuous part of the vector and $\mathbf{x}_b(k) \in \{0,1\}^{\text{n}_b}$ is the part of state vector corresponding to the discrete part. The output signals have a similar structure $\mathbf{y}(k) = [\mathbf{y}_r^{\text{T}}(k), \mathbf{y}_b^{\text{T}}(k)]^{\text{T}}$ with $\mathbf{y}_r(k) \in \mathbb{R}^{\text{m}}$ is the continuous part of the output and $\mathbf{y}_b(k) \in \{0,1\}^{\text{m}_b}$ is the discrete part of the output. $\mathbf{y}(k)$ is the output vector. The input vector $\mathbf{u}(k) = [\mathbf{u}_r^{\text{T}}(k), \mathbf{u}_b^{\text{T}}(k)]^{\text{T}}$ is composed of a continuous part $\mathbf{u}_r(k) \in \mathbb{R}^{\text{l}_r}$ and a discrete part $\mathbf{u}_b(k) \in \{0,1\}^{\text{l}_b}$. $\delta(k)$ and $\mathbf{z}(k)$ are auxiliary integer and continuous variables. Matrices $\mathbf{A}$, $\mathbf{B}_1 \sim \mathbf{B}_3$, $\mathbf{C}$, $\mathbf{D}_1 \sim \mathbf{D}_3$ and $\mathbf{E}_1 \sim \mathbf{E}_4$ denote real constant matrices, $\mathbf{E}_5$ is a real vector. This MLD formulation allows the evolution of continuous variables by linear dynamic equations, of discrete variables through described guards and the mutual interaction between them.

Here we use the hybrid system description language HYSDEL. HYSDEL is a high-level modeling language for a class of hybrid systems that is referred to as discrete hybrid automata [97]. The discrete hybrid automaton considered in our case represents a switched linear dynamic system. HYSDEL can be used for describing the discrete hybrid automaton and generating computational models from this representation (e.g., Mixed Logical Dynamical models) in an arithmetic way [5]. With HYSDEL, boolean functions for describing the transition of discrete modes can be converted to sets of inequalities and equalities involving only continuous variables. The overall continuous dynamics with associated events are denoted by a unified dynamical model with several inequalities. Using the functions defined in HYSDEL for describing discrete hybrid automata, the associated HYSDEL compiler then translates the description into a Mixed Logical Dynamical model, which can subsequently be used for system and control design. More details can be found in [97].

Considering that originally HYSDEL is not used for interconnected hybrid systems, we formulated multiple interconnected hybrid automata into a complete hybrid automaton in which all the local variables become the global variables. Then, we can use HYSDEL for translating the hybrid systems considered in this chapter. With this, the QC, the AGV and the ASC and their interactions can be described in terms of a MLD model.

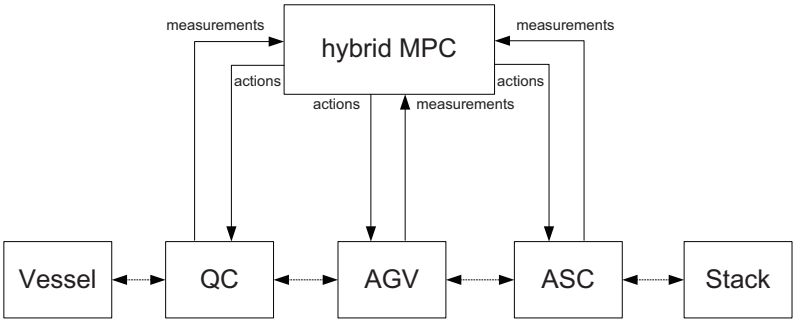


Figure 3.6: The control structure of the hybrid MPC controller.

Boundary conditions for the model (3.5)-(3.7) consist of the values for $\mathbf{x}(k)$. These need to be given in order to evaluate the MLD model. A particular geometry described using x_c^{load} and x_c^{unload} in the hybrid automata can be reflected in the MLD model. The uncertainties, e.g., the operation delay and the precise arrival time of new containers, can be incorporated based on the MLD model by measuring the states and adding new variables.

3.3 Hybrid model predictive control

MPC has been successfully used in many applications [82]. The term MPC refers to a control methodology that makes explicit use of a dynamical model to obtain control actions by minimizing an objective over a finite receding horizon. In MPC the dynamical model is used to predict the future state of the system, based on the current state and proposed future actions. These control actions are calculated by minimizing the cost function taking into account the constraints on states, outputs and inputs. MPC provides an on-line control framework for controlling systems with interacting variables, complex dynamics and constraints. In particular, MPC has been applied to hybrid systems which considers discrete-event dynamics and continuous-event dynamics together [12, 51].

We consider one centralized hybrid MPC controller for the components as shown in Fig. 3.6. The task of the controller is to transport containers from a bay of a vessel and one stack employing these components. The objective of the controller is to balance the handling capacity and the energy consumption of the controlled components. The formulated hybrid MPC problem is as follows:

$$\min \sum_{l=0}^{N_p-1} \left[J_1\left(\mathbf{x}(k+l+1), \mathbf{u}(k+l)\right) + J_2\left(\mathbf{x}(k+l+1), \mathbf{u}(k+l)\right) \right] \tag{3.8}$$

subject to, for $l = 0, 1, \ldots, N_p - 1$,

$$\mathbf{x}(k+1+l) = \mathbf{A}\mathbf{x}(k+l) + \mathbf{B}_1\mathbf{u}(k+l) + \mathbf{B}_2\delta(k+l) + \mathbf{B}_3\mathbf{z}(k+l) \tag{3.9}$$

$$\mathbf{y}(k+l) = \mathbf{C}\mathbf{x}(k+l) + \mathbf{D}_1\mathbf{u}(k+l) + \mathbf{D}_2\delta(k+l) + \mathbf{D}_3\mathbf{z}(k+l) \tag{3.10}$$

$$\mathbf{E}_2\delta(k+l) + \mathbf{E}_3\mathbf{z}(k+l) \leq \mathbf{E}_1\mathbf{u}(k+l) + \mathbf{E}_4\mathbf{x}(k+l) + \mathbf{E}_5 \tag{3.11}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(k+l) \leq \mathbf{u}_{\max} \tag{3.12}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}(k+l+1) \leq \mathbf{x}_{\max} \tag{3.13}$$
$$\mathbf{y}_{\min} \leq \mathbf{y}(k+l) \leq \mathbf{y}_{\max}, \tag{3.14}$$

where

$$J_1(\mathbf{x}(k+1),\mathbf{u}(k)) = N_{\mathrm{v}}(k) - N_{\mathrm{s}}(k) \tag{3.15}$$
$$J_2(\mathbf{x}(k+1),\mathbf{u}(k)) = \lambda_1 \left|u_{\mathrm{qc}}(k)\right| + \lambda_2 \left|u_{\mathrm{agv}}(k)\right| + \lambda_3 \left|u_{\mathrm{asc}}(k)\right|, \tag{3.16}$$

where $N_{\mathrm{v}}(k)$ represents the containers left in the vessel, $N_{\mathrm{s}}(k)$ describes the containers located in the stack. $u_{\mathrm{qc}}(k)$, $u_{\mathrm{agv}}(k)$ and $u_{\mathrm{asc}}(k)$ are the acceleration of the QC, the AGV and the ASC, respectively. It is noted that the weights for balancing the handling capacity and energy consumption are detailed using λ_1, λ_2 and λ_3 in (3.16). N_{p} is the prediction horizon, $\mathbf{x}(k+l+1)$ is the predicted state at time $k+l+1$ based on the input $\mathbf{u}(k+l)$, $\mathbf{u}_{\min}$, $\mathbf{u}_{\max}$, $\mathbf{x}_{\min}$, $\mathbf{x}_{\max}$ and $\mathbf{y}_{\min}$, $\mathbf{y}_{\max}$ are bounds on the inputs, states and outputs, respectively.

$J_1(\mathbf{x}(k+1),\mathbf{u}(k))$ addresses the handling capacity objective. The vessel is emptied as soon as possible by minimizing $N_{\mathrm{v}}(k)$, therefore the vessel can leave as soon as possible. However, $N_{\mathrm{v}}(k)$ cannot guarantee the arrival of the last container in the stack after it leaves the vessel. The term $-N_{\mathrm{s}}(k)$ is added to J_1 to guarantee the arrival of the last container to the stack.

$J_2(\mathbf{x}(k+1),\mathbf{u}(k))$ in (3.16) addresses the simplified kinetic energy consumption objective of all controlled components. Since the continuous-time dynamics represent a double-integrator, in which the air-drag and the rolling resistance are not involved, we consider the absolute value of acceleration as the cost criterion resulting from the fuel-optimal control problem of a double integrator [35]. This fuel-optimal criterion is convex and facilitates solving the optimization problem. It also removes the weight difference of the three controlled components either with a container or without a container from energy consumption representation. This simplified objective of energy consumption results in a compact formulation for implementation.

Problem (3.8)-(3.16) is a mixed integer linear programming problem (MILP). MILPs can be solved using efficient solvers such as CPLEX.

In the objective function of the proposed hybrid MPC controller, the handling part J_1 and the energy efficiency part J_2 can be balanced by altering λ_1, λ_2 and λ_3. The formulated optimization problem (3.8)-(3.16) is a multi-objective optimization problem. The tuning of λ_1, λ_2 and λ_3 for balancing J_1 and J_2 can be intensive. Also the energy consumption objective for machine (i.e., $\left|u_{\mathrm{qc}}\right|$, $\left|u_{\mathrm{agv}}\right|$ and $|u_{\mathrm{asc}}|$) is equivalent since the energy consumption objective is to sum the absolute value of accelerations over the prediction horizon for each machine. Therefore, for the sake of simplicity, we consider $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$ for the energy efficiency part J_2.

By defining $\tilde{\mathbf{u}}(k) = [\mathbf{u}(k)^{\mathrm{T}}, \mathbf{u}(k+1)^{\mathrm{T}}, \cdots, \mathbf{u}(k+N_{\mathrm{p}}-1)^{\mathrm{T}}, \delta(k)^{\mathrm{T}}, \delta(k+1)^{\mathrm{T}}, \cdots, \delta(k+N_{\mathrm{p}}-1)^{\mathrm{T}}, \mathbf{z}(k)^{\mathrm{T}}, \mathbf{z}(k+1)^{\mathrm{T}}, \cdots, \mathbf{z}(k+N_{\mathrm{p}}-1)^{\mathrm{T}}]^{\mathrm{T}}$, the standard MILP problem for the time step k can be described equivalently as follows:

$$\min_{\tilde{\mathbf{u}}(k)} \ \mathbf{f}_0^{\mathrm{T}}\tilde{\mathbf{u}}(k) + \lambda\mathbf{f}_1^{\mathrm{T}}\tilde{\mathbf{u}}(k) + \lambda\mathbf{f}_2^{\mathrm{T}}\tilde{\mathbf{u}}(k) + \lambda\mathbf{f}_3^{\mathrm{T}}\tilde{\mathbf{u}}(k) \tag{3.17}$$

subject to

$$\mathbf{b}_{\min} \leq \tilde{\mathbf{A}}\tilde{\mathbf{u}}(k) \leq \mathbf{b}_{\max} \tag{3.18}$$

$$\tilde{\mathbf{u}}_{\min} \leq \tilde{\mathbf{u}}(k) \leq \tilde{\mathbf{u}}_{\max}, \tag{3.19}$$

where $\mathbf{f}_0$ links the handling capacity in the objective function, $\mathbf{f}_1$ $\mathbf{f}_2$ and $\mathbf{f}_3$ relates the cost with respect to energy consumption of the QC, the AGV and the ASC respectively. $\mathbf{b}_{\min}$ and $\mathbf{b}_{\max}$ are the lower bound and the upper bound of the linear inequality, respectively, $\tilde{\mathbf{u}}_{\min}$ and $\tilde{\mathbf{u}}_{\max}$ are the lower bound and the upper bound of the control variables.

In the objective function (3.17), the scale of $\mathbf{f}_0^{\mathrm{T}}\tilde{\mathbf{u}}(k)$ will be significantly larger than $\mathbf{f}_1^{\mathrm{T}}\tilde{\mathbf{u}}(k)$, $\mathbf{f}_2^{\mathrm{T}}\tilde{\mathbf{u}}(k)$ and $\mathbf{f}_3^{\mathrm{T}}\tilde{\mathbf{u}}(k)$. Here we use an adaptive weight $|\mathbf{f}_0^{\mathrm{T}}\tilde{\mathbf{u}}(k-1)|$ to reduce the scale of the handling capacity part and keep the handling capacity cost in a comparably constant range. Therefore, the influence of λ on energy consumption and the handling capacity can be seen more clearly using the adaptive weight. The new objective function is given as follows:

$$\min_{\tilde{\mathbf{u}}(k)} \frac{\mathbf{f}_0^{\mathrm{T}}\tilde{\mathbf{u}}(k)}{|\mathbf{f}_0^{\mathrm{T}}\tilde{\mathbf{u}}(k-1)+\varepsilon|} + \lambda(\mathbf{f}_1^{\mathrm{T}}\tilde{\mathbf{u}}(k) + \mathbf{f}_2^{\mathrm{T}}\tilde{\mathbf{u}}(k) + \mathbf{f}_3^{\mathrm{T}}\tilde{\mathbf{u}}(k)), \tag{3.20}$$

where ε is a very small number in case $\mathbf{f}_0^{\mathrm{T}}\tilde{\mathbf{u}}(k-1) = 0$.

In the formulated objective function (3.20), the penalty λ can influence the handling capacity and the energy consumption of the pieces of equipment.

3.3.1 Extension for the external input

When the arrival of new containers is considered, the system model needs to be changed by taking into the number of new containers as an input in the dynamical model of the container handling system. The new model extends the MLD model (3.5), (3.6) and (3.7) as follows:

$$\begin{aligned}
&\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\mathbf{u}(k) + \mathbf{B}_2\delta(k) + \mathbf{B}_3\mathbf{z}(k) + \mathbf{B}_4\mathbf{d}(k)\\
&\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}_1\mathbf{u}(k) + \mathbf{D}_2\delta(k) + \mathbf{D}_3\mathbf{z}(k) + \mathbf{D}_4\mathbf{d}(k)\\
&\mathbf{E}_2\delta(k) + \mathbf{E}_3\mathbf{z}(k) \leq \mathbf{E}_1\mathbf{u}(k) + \mathbf{E}_4\mathbf{x}(k) + \mathbf{E}_5,
\end{aligned} \tag{3.21}$$

where $\mathbf{d}(k)$ is the term that indicates the number of new containers arriving, $\mathbf{B}_4$ and $\mathbf{D}_4$ reflect its influence on the state and the output.

The extended model incorporates the new arrival of containers as the external input to change the number of containers in the vessel over time and further influence the state of the MLD model, e.g. the position and the velocity of the machines. The individual dynamics of the machines do not change in the extend MLD model, but the choice of actions of the machines indeed is changed by that. In order to evaluate the MLD model at a particular time step, the value of d(k) needs to be given.

3.3.2 Performance indicators

In the objective function of the proposed MPC controller, we formulate two parts: the handling capacity part J_1 and the energy efficiency part J_2. For the compact container terminal, the handling capacity is reflected as the completion time of the transport of all containers while the energy consumption referred to the energy consumed for transpoting the containers. Therefore, we use the completion time (KPI 2) and the energy consumption (KPI 3) as the main performance indicators to evaluate the effect of the proposed MPC controller.

The calculation of the completion time of all containers is given as follows:

$$k_{\text{finish}} = \min k, \text{ subject to } N_{\text{s}}(k) = N \tag{3.22}$$

Table 3.3: The weight parameters of the controlled components.

$m_{\text{qc}}^{\text{unload}}$	$m_{\text{qc}}^{\text{load}}$	$m_{\text{agv}}^{\text{unload}}$	$m_{\text{agv}}^{\text{load}}$	$m_{\text{asc}}^{\text{unload}}$	$m_{\text{asc}}^{\text{load}}$
10 ton	25 ton	15 ton	30 ton	240 ton	255 ton

The energy consumption of all machines is formulated as follows:

$$E_{\text{tot}} = E_{\text{qc}} + E_{\text{agv}} + E_{\text{asc}}, \tag{3.23}$$

where E_{tot} describes the energy consumption of all machines, E_{qc}, E_{agv} and E_{asc} denote the energy consumption of the QC, the AGV and the ASC, respectively.

The calculation of E_{qc} is presented here:

$$E_{\text{qc}} = \sum_{k=0}^{N_{\text{sim}}-1} E_{\text{qc}}(k) \tag{3.24}$$

$$E_{\text{qc}}(k) = \begin{cases} 0.5 m_{\text{qc}}(k) \times \\ (v_{\text{qc}}^2(k+1) - v_{\text{qc}}^2(k)) & v_{\text{qc}}^2(k+1) > v_{\text{qc}}^2(k) \\ 0 & \text{else} \end{cases} \tag{3.25}$$

$$m_{\text{qc}}(k) = \begin{cases} m_{\text{qc}}^{\text{unload}} & v_{\text{qc}}(k+1) < 0 \\ m_{\text{qc}}^{\text{load}} & v_{\text{qc}}(k+1) > 0, \end{cases} \tag{3.26}$$

where N_{sim} is the simulation length, $m_{\text{qc}}^{\text{unload}}$, $m_{\text{agv}}^{\text{unload}}$, $m_{\text{asc}}^{\text{unload}}$is the unloading weight of the QC, the AGV and the ASC, respectively, without the container. $m_{\text{qc}}^{\text{load}}$, $m_{\text{agv}}^{\text{load}}$, $m_{\text{asc}}^{\text{load}}$ is the loading weight of the QC, the AGV and the ASC, respectively, with the container. The weights of the machines [41, 42, 54] are given in Table 5.4. E_{agv} and E_{asc} can be computed in a similar way as E_{qc}.

Besides the completion time (KPI 2) and the energy consumption (KPI 3), the computation time (KPI 4) and the utilization of the QC, the AGV and the ASC (KPI 7-9) are used for evaluating the performance of the proposed controller additionally.

3.4 Simulation experiments

In this section, the choice of the prediction horizon and the trade-off between the handling capacity and the energy consumption of the proposed hybrid MPC controller will be first discussed. Afterwards, the adaptiveness and the robustness of the hybrid MPC controller will be presented in comparison to the discrete-event based distributed controller (DDC).

3.4.1 Prediction horizon choice

To search for a proper N_{p}, we test the effect of handling all containers by changing the N_{p} of the proposed MPC controller. We define the value '1' when all containers can be handled

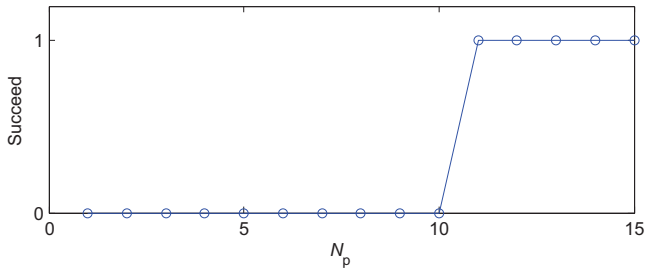


Figure 3.7: The effect of simulation with respect to the prediction horizon N_p.

completely, and '0' if all containers cannot be transported to the stack. To test different N_p, the number of containers N is assumed to be 10 in this simulation.

Fig. 3.7 indicates that the short prediction horizon cannot guarantee all containers be transported from a bay in the vessel to the storage place in the stack. Short prediction horizons cannot predict the complete interaction of all pieces of machines. Due to the incomplete prediction, the pieces of equipment stay at one place. Therefore, all containers cannot be transported to the destination with a short prediction horizon.

In the example presented in Fig. 3.7, a value larger than 10 time steps can be chosen as N_p for this particular layout. However, for a different layout, the value chosen based on Fig. 3.7 may not be sufficient to guarantee the transport of all containers. This is because a longer transport time can be possible when a different layout is given, resulting in a longer prediction horizon. In particular, a heavy penalty on energy consumption can result in slow transport of a container. Under such circumstances a larger N_p is required for guaranteeing the transport of all containers, at the cost of increased computation time. Therefore, we choose $N_p = 25$ for the prediction horizon of the proposed hybrid MPC controller. The chosen N_p could be suitable for different terminal handling layouts in which a sufficient prediction is possibly required for the complete prediction of the interactions of the pieces of equipment.

3.4.2 Balancing handling capacity and energy consumption

In the formulated objective function (3.20), the penalty λ can influence the handling capacity and the energy consumption of the pieces of equipment. We vary λ to illustrate the trade-off between the completion time t_{finish} and total energy consumption E_{tot}.

The influence of the parameter λ on energy consumption and the completion time for the case of transporting 5 containers is shown in Fig. 3.8. The figure illustrates that generally a higher λ can reduce energy consumption, but increase the completion time of the container handling system. Due to the high penalty on the cost for energy consumption, the machine slows down the process of acceleration and deceleration and a low velocity is obtained. Therefore, the kinetic energy consumption is reduced. It is also noticed that a smaller penalty λ can result in more energy-efficient handling system. The energy efficiency of the controller maintains the minimal completion time, compared with the MPC controller without penalty on energy consumption. Based on the simulation results, we choose $\lambda = 0.05$ for the energy-efficient MPC controller. For the sake of notation, we ab-

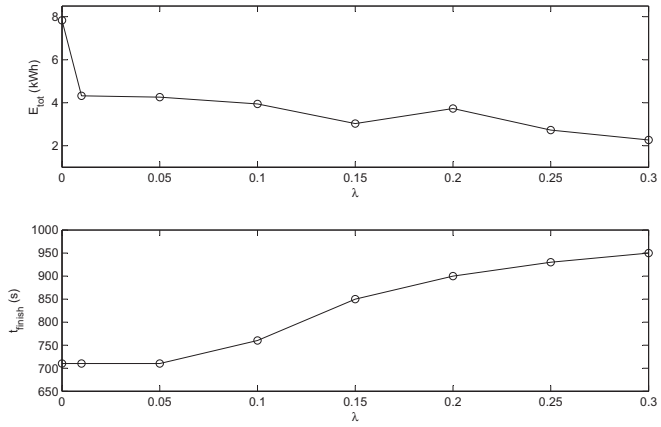


Figure 3.8: The simulation results of the total energy consumption and finishing time for varying λ when 5 containers are transported.

Table 3.4: Three container handling layouts.

	layout 1	layout 2	layout 3
d_{qc}	80m	80m	120m
d_{agv}	120m	200m	100m
d_{asc}	100m	100m	80m

breviate the energy-efficient hybrid MPC with $\lambda = 0.05$ as EHMPC in the following part.

3.4.3 Adaptiveness

Above we use one layout of the container handling system to test the EHMPC controller for transporting containers. Still, in practice the layout of the container handling system can vary according to the size of vessel and the different storage place in the stack.

For a given layout of three machines, the transfer of a containers requires that two interacting machines must be available. The AGV can leave the quayside with a container only when the QC is available with a new container. At the same time, the QC can transfer a container only when the empty AGV is available. A similar interaction also happens when it comes to the AGV and the ASC. Therefore, one of the QC, the AGV and the ASC can become the bottleneck when it is expected to arrive when another machine is waiting.

Here we consider three cases of terminal layouts in which the bottleneck of each layout is different parameters. The parameters of these three layouts are given in Table 3.4. Parameters d_{qc}, d_{agv} and d_{asc} are the traveling distance of the QC, the AGV and the ASC, respectively. These parameters are chosen based on a typical container terminal. The robustness of the EHMPC to different layouts can be tested with respect to different bottlenecks. We test the transport of 5 containers and 10 containers to illustrate that EHMPC controller can be applied to different layouts of a container terminal.

Table 3.5: The performance of controllers with respect to layout 1.

container	5			10		
controller	DDC	EHMPC	change	DDC	EHMPC	change
t_{finish}	700s	700s	0	1350s	1350s	0
E_{qc}	0.37kWh	0.25kWh	-32%	0.76kWh	0.42kWh	-44%
E_{agv}	1.13kWh	0.56kWh	-50%	2.25kWh	1.15kWh	-49%
E_{asc}	3.81kWh	3.45kWh	-9%	7.64kWh	7.27kWh	-5%
E_{tot}	5.31kWh	4.26kWh	-20%	10.64kWh	8.84kWh	-17%
QC utilization	90%	100%	+10%	87%	100%	+13%
AGV utilization	87%	100%	+13%	86%	100%	+14%
ASC utilization	100%	100%	0	100%	100%	0

Table 3.6: The performance of controllers with respect to layout 2.

container	5			10		
controller	DDC	EHMPC	change	DDC	EHMPC	change
t_{finish}	840s	840s	0	1610s	1610s	0
E_{qc}	0.37kWh	0.17kWh	-54%	0.76kWh	0.31kWh	-59%
E_{agv}	0.78kWh	0.73kWh	-8%	1.56kWh	1.51kWh	-3%
E_{asc}	3.82kWh	2.49kWh	-35%	7.64kWh	5.24kWh	-31%
E_{tot}	4.97kWh	3.39kWh	-31%	9.96kWh	7.06kWh	-29%
QC utilization	89%	100%	+11%	88%	100%	+12%
AGV utilization	100%	100%	0	100%	100%	0
ASC utilization	89%	100%	+11%	88%	100%	+12%

The results with respect to the performance of controllers in three layouts are presented in Table 3.5, Table 3.6 and Table 3.7.

In general, in Table 3.5, Table 3.6 and Table 3.7 we observe that energy consumption is reduced consistently by EHMPC in three different layouts of the container terminal handling system, compared with the DDC controller. For these three layouts, the average energy consumption is reduced by 27 % for the same completion time. As shown in Table 3.5, the total energy consumption of transporting 5 containers and 10 containers decrease by 20% and 17% in the layout 1. The total energy consumption of transporting 5 containers and 10 containers in the layout 2 are reduced by 31% and 29%, which is illustrated in Table 3.6. Table 3.7 shows the total energy consumption of transporting 5 containers and 10 containers in the layout 2 are reduced by 35% and 33%. The total energy reduction in the layout 2 and in the layout 3 are more significant than it in the layout 1. As shown in Table 3.5, the utilization of the ASC is 100% and therefore the ASC is the bottleneck of the layout 1, maintaining the fast transport of containers. Therefore, the ASC has to move at its maximal speed, resulting in high energy consumption.

Table 3.7: The performance of controllers with respect to layout 3.

container	5			10		
controller	DDC	EHMPC	change	DDC	EHMPC	change
t_{finish}	700s	700s	0	1610s	1610s	0
E_{qc}	0.37kWh	0.37kWh	0	0.76kWh	0.76kWh	0
E_{agv}	0.78kWh	0.57kWh	-27%	1.56kWh	1.20kWh	-23%
E_{asc}	5.50kWh	3.36kWh	-39%	11.00kWh	7.03kWh	-36%
E_{tot}	6.65kWh	4.30kWh	-35%	13.32kWh	8.99kWh	-33%
QC utilization	100%	100%	0	100%	100%	0
AGV utilization	87%	100%	+13%	86%	100%	+14%
ASC utilization	87%	100%	+13%	86%	100%	+14%

3.4.4 Robustness

Section 4.4 discussed the robustness of the EHMPC controller when it is applied to different layouts. In the simulated scenarios, all the containers are transported as planned without any uncertainties. However, uncertainties may take place in the container handling system, e.g., the delay of operation and the precise time at which new containers arrive. These two types of uncertainties will be investigated next.

Operation delay

In the handling process of a container terminal, one machine may have a delay resulting from the handling operation (e.g., unload or load a container). This requests that the EHMPC controller can adapt the delayed condition and transport the remaining containers still in an energy-efficient way. In this chapter, we simulate an operation delay of 60s occurring when the QC picks up a container from the vessel since the processed container can not be grabbed properly. Due to the delay of the QC, the trajectories of the other two machines have to be re-optimized for the sake of energy-efficient handling. On the one hand the complete time needs to be minimized, while the energy reduction is sill expected on the other hand. To test the performance of the EHMPC controller, we set up a scenario in which a group of containers is transported from the vessel. In this scenario, the number of containers is assumed to be 5 and a 60-second delay takes place when the third container is unloaded by the QC. This scenario is simulated both for the DDC controller and the EHMPC controller.

The trajectories of three machines determined by the DDC controller are presented in Fig. 3.9, Fig. 3.11 and Fig. 3.13, while the trajectories of three machines determined by the EHMPC controller are given in Fig. 3.10, Fig. 3.12 and Fig. 3.14 as comparison. The moment at which the delay takes place is indicated in the straight line in these figures.

Fig. 3.9, Fig. 3.11 and Fig. 3.13 show the fixed moving behavior of the ASC and the ASC when the DDC controller is implemented. The AGV and the ASC slows down to reduce energy when the delay of the QC starts but later speed up again when the delay finishes.

Fig. 3.15 presents the computation time of the EHMPC controller at step k. It is seen that the computation time is consistent with the transport of containers, shown in Fig. 3.10, Fig. 3.12 and Fig. 3.14. The computation time reduces when the container transport is going

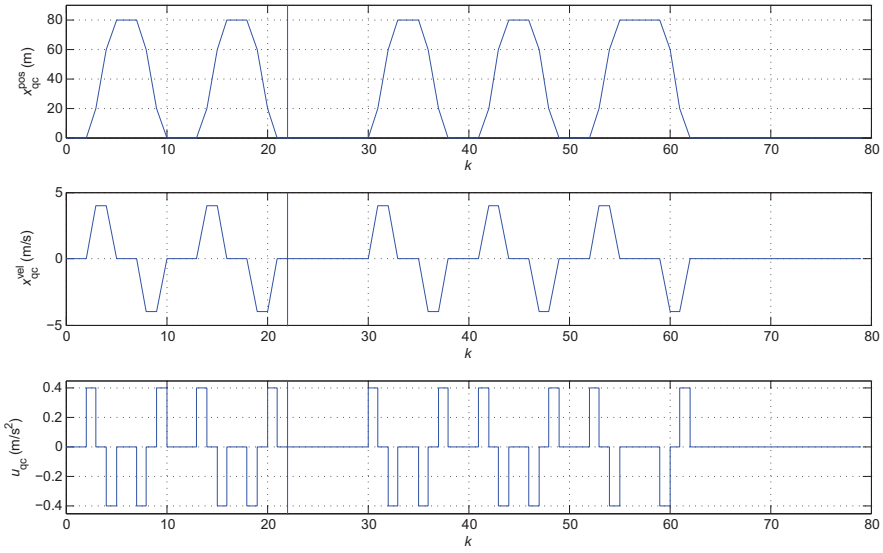


Figure 3.9: The trajectories of the QC controlled by DDC in the scenario of operation delay.

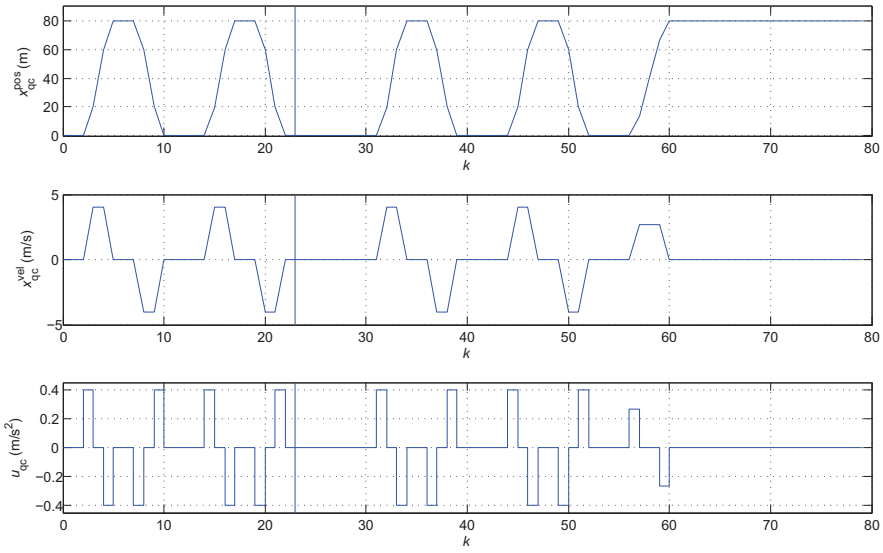


Figure 3.10: The trajectories of the QC controlled by HEMPC in the scenario of operation delay.

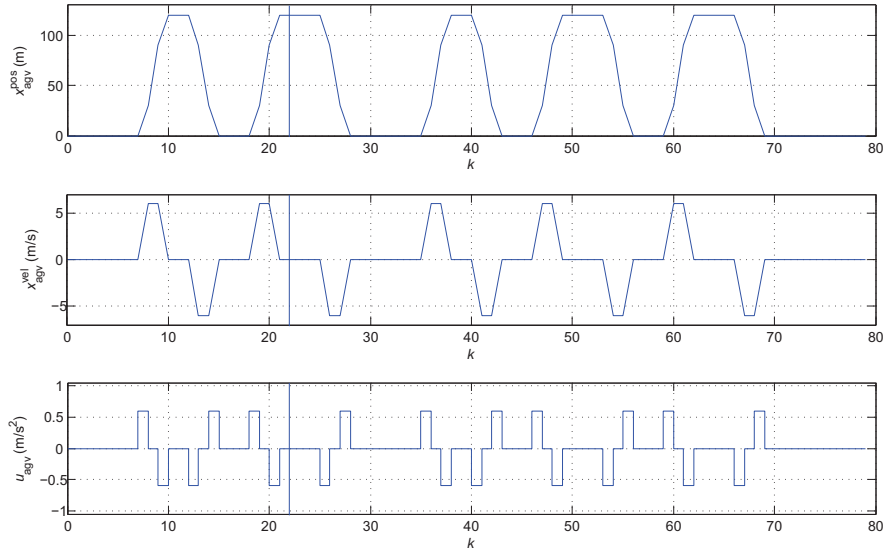


Figure 3.11: The trajectories of the AGV controlled by DDC in the scenario of operation delay.

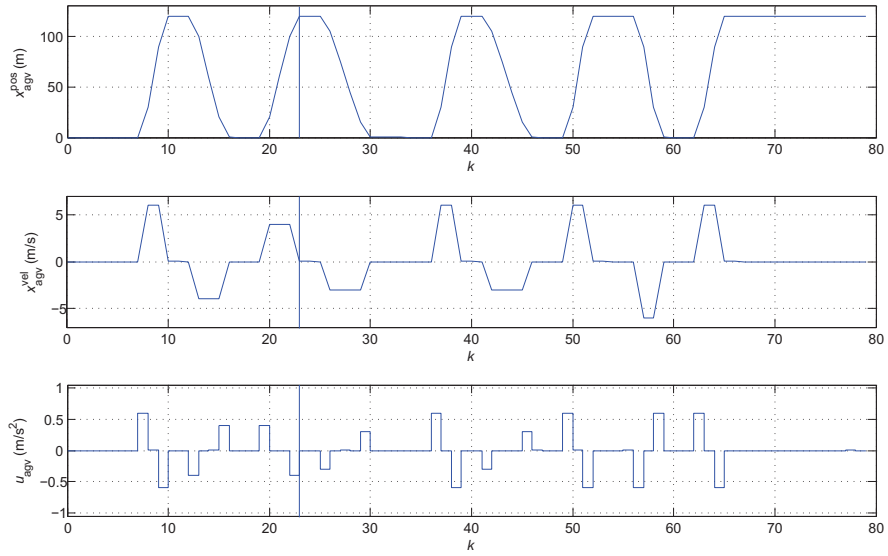


Figure 3.12: The trajectories of the AGV controlled by HEMPC in the scenario of operation delay.

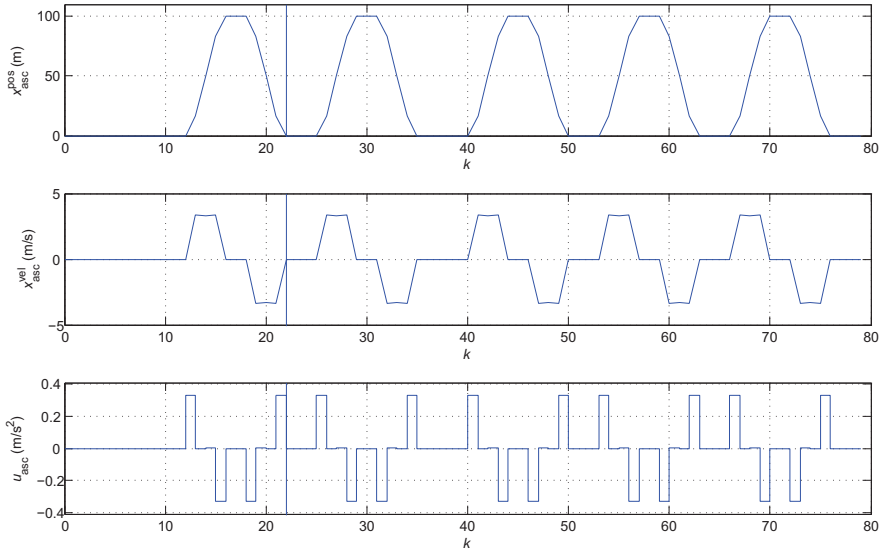


Figure 3.13: The trajectories of the ASC controlled by DDC in the scenario of operation delay.

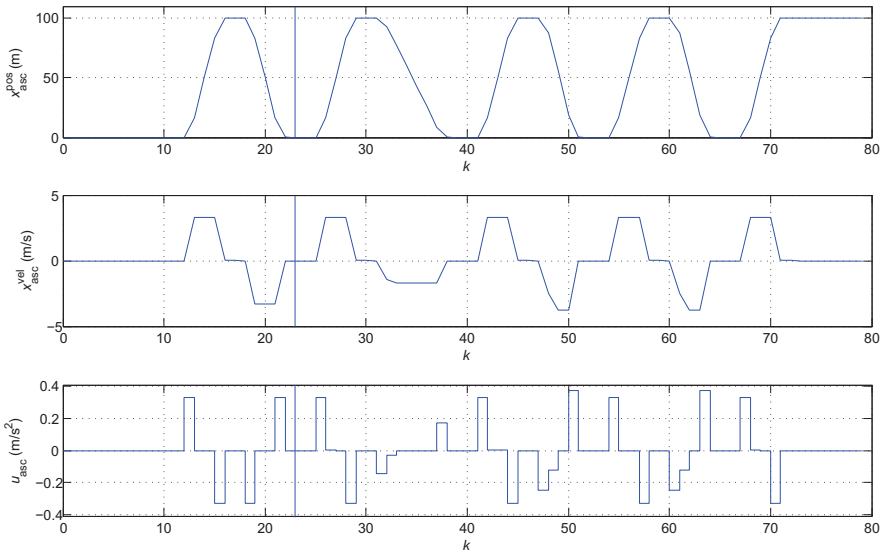


Figure 3.14: The trajectories of the ASC controlled by HEMPC in the scenario of operation delay.

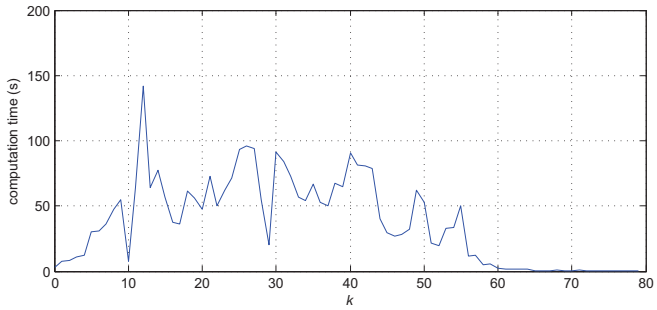


Figure 3.15: The computation time of EHMPC at step k in the scenario of operation delay.

Table 3.8: The energy consumption of machines with respect to the layout 1 in the scenario of operation delay.

container	5 (with a delay)		
controller	DDC	EHMPC	change
t_{finish}	730s	740s	+1 %
E_{qc}	0.39kWh	0.33kWh	-15 %
E_{agv}	1.13kWh	0.81kWh	-28 %
E_{asc}	3.81kWh	3.37kWh	-12 %
E_{tot}	5.33kWh	4.51kWh	-15 %

to finish correspondingly.

Table 5.7 compares the performance of different controllers in the scenario of operation delay. In the case of 60s delay, there is still 15 % of energy reduction from the EHMPC controller than the DDC controller. Both for the EHMPC controller, compared to the result in Table 3.5 without the delay, in Table 5.7 the AGV and the ASC use more energy to speed up the transport of containers after the delay stops. One can observe that the completion time of EHMPC is 10s later than that of the DDC. This is because the QC arrives at the vessel 10s later when EHMPC is used than when DDC is employed.

Arrival of new containers

Besides the uncertainty resulting from the operation delay, the exact time at which new containers arrive is regarded as another uncertainty that influences the real-time control of container handling. To test the performance of the proposed EHMPC, we set up a scenario in which another group of containers arrives as a known disturbance. In this scenario, we assume 3 containers will be handled as planned and another 2 containers arrives later for illustrating how this disturbance is handled by the EHMPC controller.

In the scenario of the new arrival containers, the trajectories of three machines determined by the DDC controller are presented in Fig. 3.16, Fig. 3.18 and Fig. 3.20, while the trajectories of three machines determined by the EHMPC controller are given in Fig. 3.17, Fig. 3.19 and Fig. 3.21 as comparison. The moment at which the new containers arrive is

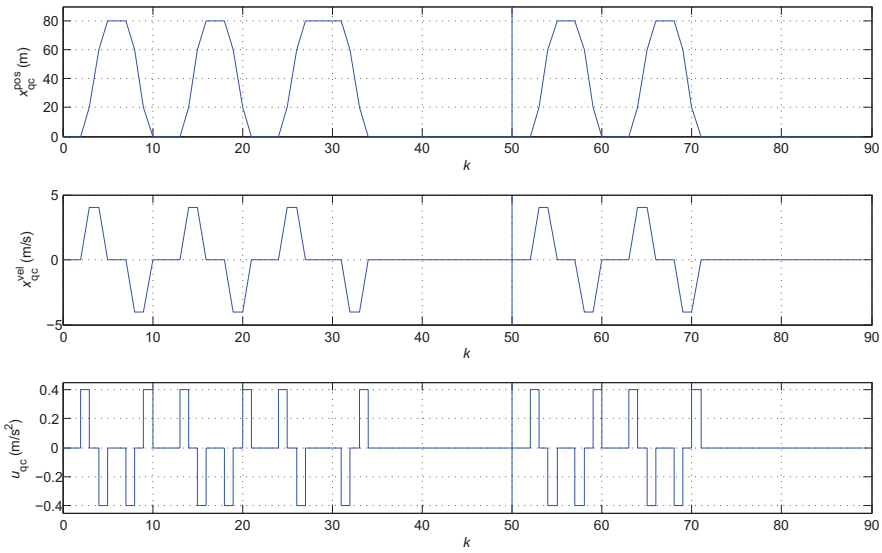


Figure 3.16: The trajectories of the QC controlled by DDC when new containers arrives.

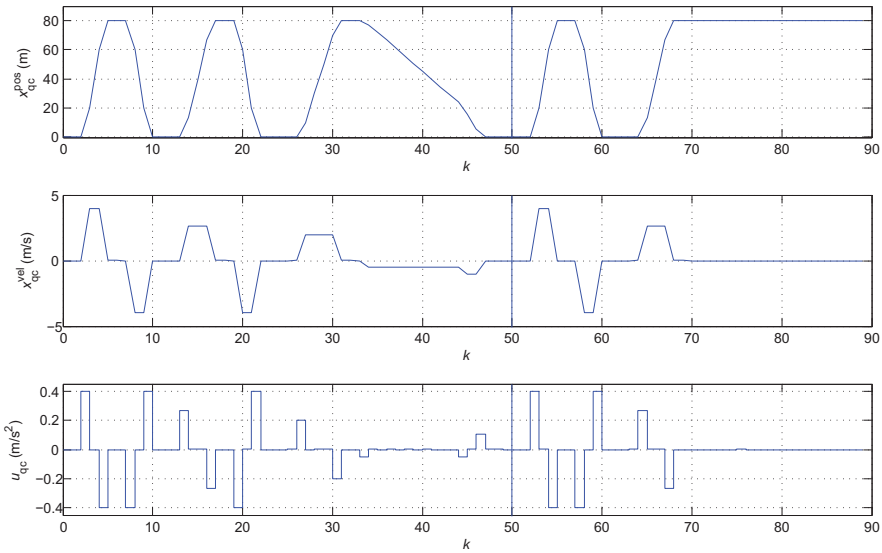


Figure 3.17: The trajectories of the QC controlled by EHMPC when new containers arrives.

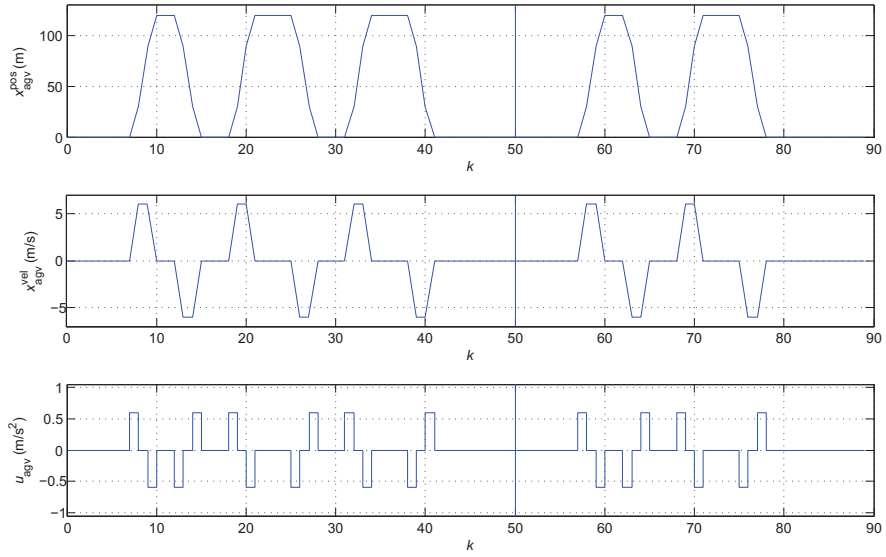


Figure 3.18: The trajectories of the AGV controlled by DDC when new containers arrives.

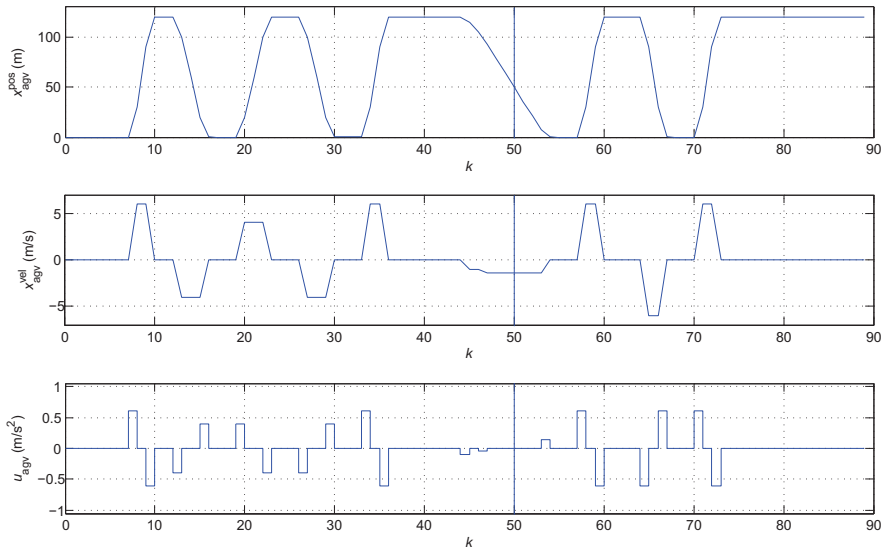


Figure 3.19: The trajectories of the AGV controlled by EHMPC when new containers arrives.

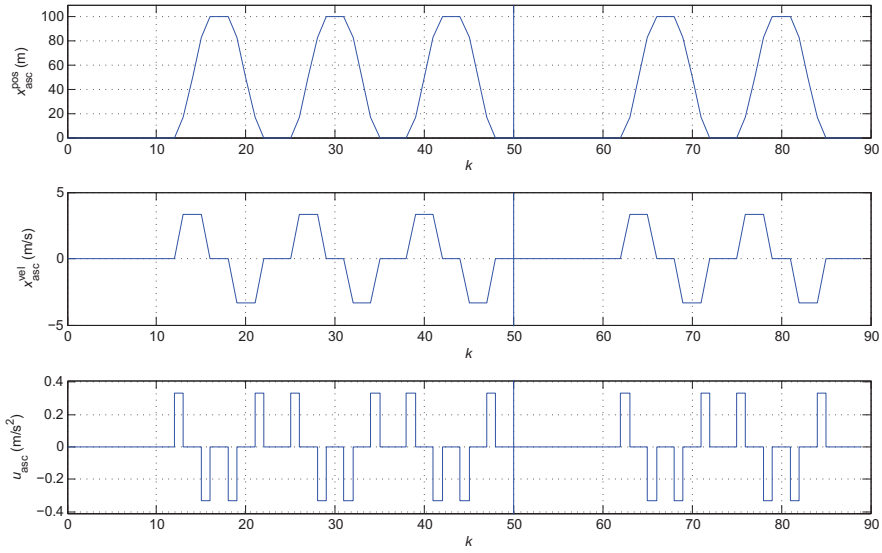


Figure 3.20: The trajectories of the ASC controlled by DDC when new containers arrives.

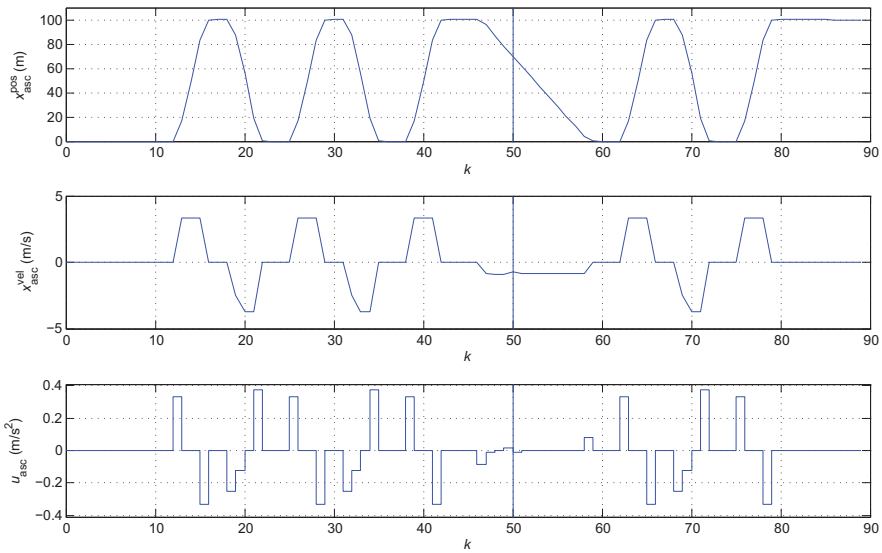


Figure 3.21: The trajectories of the ASC controlled by EHMPC when new containers arrives.

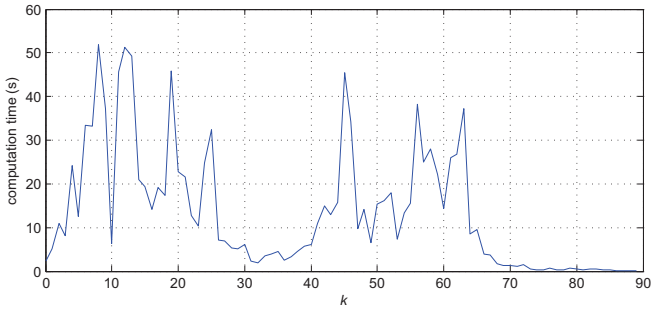


Figure 3.22: The computation time of EHMPC at step k in the scenario of new arrival containers.

Table 3.9: The energy consumption of equipment with respect to the layout 1.

container	3 containers + 2 new containers		
controller	DDC	EHMPC	change
t_{finish}	820s	820s	0%
E_{qc}	0.39kWh	0.24kWh	-38%
E_{agv}	1.13kWh	0.81kWh	-28%
E_{asc}	3.81kWh	3.40kWh	-11%
E_{tot}	5.31kWh	4.45kWh	-16%

indicated in the straight line in these figures.

Fig. 3.16, Fig. 3.18 and Fig. 3.20 show the fixed moving behavior of the ASC and the ASC when the DDC controller is implemented. It is seen from Fig. 3.17, Fig. 3.19 and Fig. 3.21 that the QC, the AGV and the ASC move slowly before the arrival of new containers to reduce energy consumption.

Fig. 3.22 presents the computation time of the EHMPC controller at step k. It is seen that the computation time is consistent with the transport of containers, shown in Fig. 3.17, Fig. 3.19 and Fig. 3.21. The computation time decreases when the container transport is going to finish.

Table 3.9 compares the performance of the DDC controller and the EHMPC controller when it comes to the arrival of new containers. The EHMPC controller gains 12% of energy reduction in total, compared to the DDC controller. The energy consumed by the QC, the AGV and the ASC all decrease when the EHMPC controller takes the information of the new arrival containers into account. For energy efficiency, the EHMPC controller does not increase the completion time, as compared to the DDC controller.

3.5 Concluding remarks

In this chapter, a control structure for the bay handling task integrating the scheduling and control is proposed. For this, the continuous-time and discrete-event dynamics of five interacting components in a container handling system are modeled by hybrid automata. The

hybrid system considered in this chapter is transformed into a mixed logical dynamical (MLD) model for the control purpose using HYSDEL. A hybrid model predictive control (MPC) controller for these machines is then proposed to balance the handling capacity and energy consumption of machines. A proper prediction horizon N_p is chosen to incorporate complete interactions of different machines. The trade-off between the energy consumption E_{tot} and the completion time t_{finish} is illustrated in a simulation. To maintain the maximal handling capacity in an energy-efficient way, a small penalty λ is chosen in the objective function of MPC. The simulations indicate that the proposed controller obtains energy efficiency when different layouts of terminal handling systems are considered, in which the average energy consumption is reduced by 27 % for the same completion time. The simulations also show the controller can handle two types of uncertainties in real-time container handling.

Next, in Chapter 4 energy efficiency of a medium-size container terminal will be investigated. As the system scale increases, the discrete-event dynamics include the orders of containers processed by the piece of equipment and the assignment of containers to the particular pieces of equipment, resulting in a more complex control problem than the compact terminal. For the medium-size terminal, using the hybrid MPC approach the binary decision variables increase considerably and solving this control problem using the hybrid MPC approach is intractable. Therefore the approach of this chapter can not then be used. Chapter 4 will deal with this complex control problem.

Chapter 4

Energy-aware control using integrated flow shop scheduling and optimal control

Chapter 3 discussed the energy efficiency of the operations of the compact container terminal, wherein a hybrid MPC controller is proposed for real-time operations. This chapter focuses on the energy efficiency of the medium-size container terminal. The operation of the medium-size terminal involves more complex discrete-event dynamics than the compact terminal, increasing the difficulty for controlling the pieces of equipment. To tackle this problem, a new methodology will be proposed.

The research discussed in this chapter is based on [109, 110].

4.1 Introduction

In this chapter, energy efficiency of the medium-size container terminal is investigated. The medium-size terminal has higher complexity than the compact one for controlling the pieces of equipment when containers are transported. The compact terminal considers the case of one QC, one AGV and one ASC. The assignment of containers to equipment and sequence in which jobs are processed by pieces of equipment are then straightforward. When it comes to the medium-size container terminal, the number of pieces of equipment increases and the computational complexity for controlling the employed pieces of equipment grows correspondingly. The medium-size container terminal considers the case of one QC, multiple AGVs and multiple ASCs. As a whole, these pieces of equipment are operated cooperatively in order to optimize the handling of containers in an energy-efficient way. In this situation, determining the sequence in which jobs are processed by pieces of equipment is not straightforward. Also, the assignment problem in which a container is assigned to a particular piece of AGVs and ASCs is considered in the overall problem. To cope with the complexity of controlling all employed pieces of equipment, an appropriate control system has to be designed.

To reduce complexity for controlling the medium-size container terminal, a hierarchical

control architecture is proposed in this chapter. The behavior of the medium-size terminal is considered as consisting of a higher level and a lower level represented by discrete-event dynamics and continuous-time dynamics, respectively. These dynamics represent the behavior of a large number of terminal equipment. For controlling the higher level dynamics, a hybrid flow shop scheduling problem for the minimal completion time is solved. In this chapter, the minimal completion time is referred to as makespan for the scheduling problem, which is typically defined as the completion time of all jobs in the filed of operations research. For this, the minimal time required by a particular piece of equipment for performing an operation at the lower level is needed. As an analytical solution to this optimal control problem, Pontryagins Minimum Principle is used for calculating the minimal time for performing an operation at the lower level. The actual operation time allowed by the higher level for processing an operation at the lower level is subsequently determined by an energy-efficient scheduling algorithm at the higher level. Given an actual operation time, the lower level dynamics are controlled using optimal control to achieve minimal energy consumption while respecting the time constraint. Simulation studies illustrate how energy-efficient management of equipment for the minimal makespan could be obtained using the proposed methodology.

The remainder of this chapter is organized as follows: Section 4.2 presents the decomposition of the dynamics of automated container terminals involving three types of equipment. Section 4.3 subsequently proposes a hierarchical architecture for controlling the equipment. Section 4.4 illustrates the potential of the proposed approach in multiple simulation studies using the benchmark system. Section 4.5 concludes this chapter.

4.2 Modeling of container handling equipment

As shown in Fig. 4.1, the medium container terminal consists of one QC, multiple AGVs and multiple ASCs. In a typical unloading cycle, a QC picks up a container from the vessel and then unloads it to an AGV. The AGV moves with the container from the quayside to the stacking area, where a container is unloaded by an ASC. The ASC then transports the container to the position in the storage area. In a loading cycle these movements are reversed. Accelerations, decelerations and steering angles (if applicable) of the pieces of equipment have to be determined in an optimal way. In addition, the moment at which containers are transported from one piece of equipment to the next will be determined. In this chapter, for the sake of simplicity only the horizontal trajectory of equipment is considered. The trajectory of AGVs is simplified into one-dimension movements and the steering angle of AGVs is assumed fixed to simplify the modeling. In Chapter 6, a higher order dynamics will be considered. In the case of one QC, the number of AGVs is small and therefore collision avoidance of AGVs is not considered in this chapter. Collision avoidance of AGVs will be considered explicitly in the case of multiple QCs as discussed in Chapter 6. The AGVs are assumed to be dynamically identical.

The dynamics of the pieces of equipment considered are driven by discrete events when a container is transferred from one piece of equipment to another one. Meanwhile, the continuous dynamics, i.e., the position and the speed of a piece of equipment, evolve between these discrete changes. The dynamics of transporting containers can therefore be represented by the combination of discrete-event dynamics and continuous-time dynamics. The

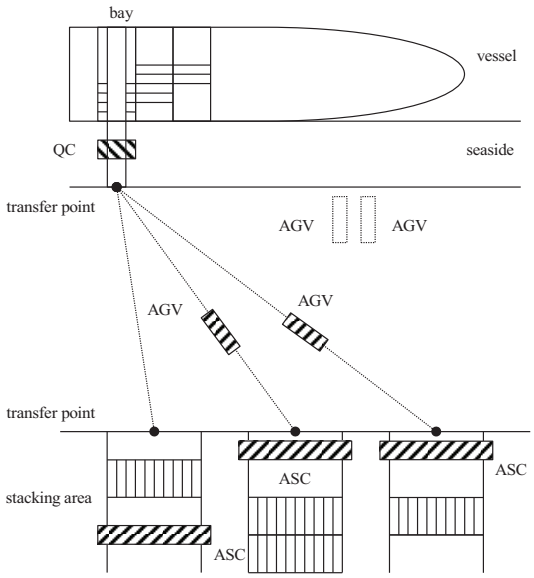


Figure 4.1: Schematic layout of equipment in a medium container terminal.

vessel and the stacking area are hereby considered as boundary components that have no internal dynamics.

4.2.1 Hierarchical decomposition

In general, a large class of hybrid systems can be described by the architecture of Fig. 4.2 [39]. In this architecture a typical hybrid system is arranged into two layers. Different levels of abstraction of the system model are used at each level of the hierarchy. At the lower level, the system model is usually described by means of differential-algebraic or difference equations. At the higher level the system description is more abstract using discrete-event modeling. Based on the decomposition of system dynamics, a control architecture can be build. Typically the controller designed for the top level is then a discrete event supervisory controller (see, e.g., [81]), while the lower-level controller is controlling the continuous-time dynamics. The higher level and the lower level communicate by means of an interface that translates continuous signals and discrete events. On the one hand, the discrete state of the higher layer triggers the continuous-time dynamics in the lower layer. On the other hand, the signal of the lower layer generates an event to drive the dynamics of the higher layer. For most of these systems the control design approach has been “divide and conquer”, i.e., the higher-level and the lower-level controllers are designed independently and then linked by an interface that is designed for the specific problem. In this chapter we apply this decomposition to reduce the control complexity of an automated container terminal.

The dynamics of transporting containers as considered in this chapter can also be decomposed into the two levels. For the high-level discrete-events dynamics, the models (Finite State Machines, Petri Nets, Max-plus Equations) in Fig. 4.2 resulting from control theory are not direct for modeling of the pieces of equipment and their interactions. The hybrid

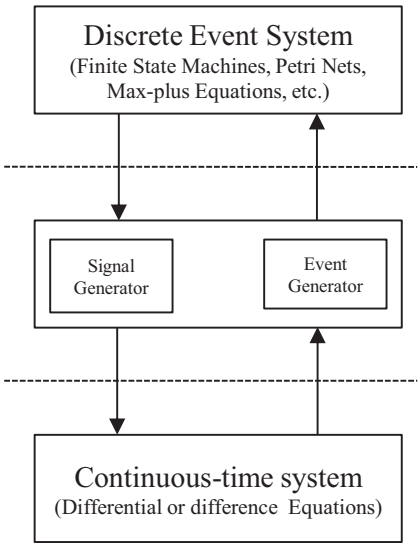


Figure 4.2: System dynamics decomposed into a discrete-event and a continuous-time part (based on [6, 39, 40]).

flow shop [86] is typically used in operations research and logistics and we hereby propose it as a new model for this framework. This model can describe the exchanging of a container by two types of equipment. The lower-level continuous-time dynamics can be modeled as differential equations that describe the dynamics of one piece of equipment for transporting a container. As will be detailed below, these two levels can be linked by the operation time allowed for each piece of equipment for doing a certain operation. The higher-level models can subsequently be used for scheduling of the discrete-event interactions of equipment; the lower-level model can be used for control of each individual piece of equipment. Below the discrete-event model for the interaction of equipment and the continuous-time model for each individual piece of equipment will be discussed in detail.

4.2.2 Higher-level discrete-event dynamics

At the higher level of the system the behavior is represented by discrete-event dynamics. The discrete-event dynamics describe within which time interval and in which sequence a number of containers is handled by the available pieces of equipment (i.e., QC, AGV, ASC).

The operations of the pieces of equipment can be represented as a particular discrete-event system, referred to as three-stage hybrid flow shop [86]. In a hybrid flow shop, each job has to pass through a number of stages. At every stage a number of identical machines can be used in parallel to process the jobs. A job is processed by the sequence of stage and each job requires a certain processing time in each stage.

In the three-stage flow shop that we consider, a job is defined as a complete process in which a container is transported from the vessel to its stacking position. One job is processed by three types of equipment (QC, AGV and ASC). As a three-stage hybrid flow shop, the operations by the three types of equipment are described in terms of three stages:

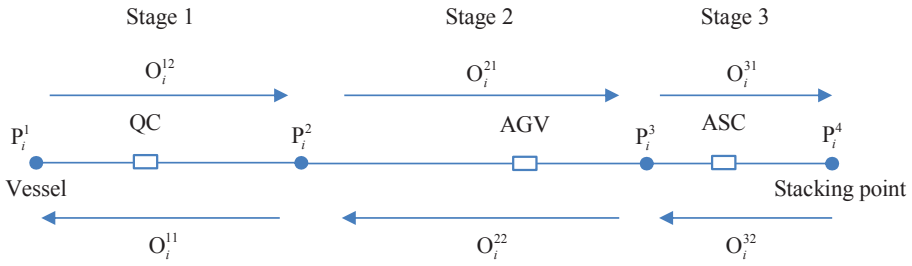


Figure 4.3: The sequence of transporting containers using three pieces of equipment. A job involves going through each of the operations.

1. Stage 1: transport by the QC
2. Stage 2: transport by one of the AGVs
3. Stage 3: transport by one of the ASCs

The operations by the three types of equipment are illustrated in Fig. 4.3. P_i^1 is defined as the place of container i in the vessel. P_i^2 is the defined as the transfer point at which container i is transferred from a QC to an AGV. P_i^3 is defined as the transfer point at which container i is transferred from an AGV to an ASC. P_i^4 is defined as the storage place of container i in the stack.

In stage 1, there are two operations O_i^{11} and O_i^{12}. Operation O_i^{11} is defined as the move of the QC from P_i^2 to P_i^1 for container i and operation O_i^{12} is defined as the move of the QC from P_i^1 to P_i^2 with container i. In stage 2, there are two operations O_i^{21} and O_i^{22} in which an AGV moves from P_i^2 to P_i^3 with container i and the AGV returns from P_i^3 to P_i^2 after unloading container i, respectively. Two operations O_i^{31} and O_i^{32} are defined in stage 3, in which an ASC transports container i from P_i^3 to P_i^4 and the ASC moves from P_i^4 to P_i^3 after unloading container i, respectively.

We next define the hybrid flow shop *problem* for our situation. The hybrid flow shop problem consists of finding the sequences of jobs and equipment handling these jobs in an optimal way. In order to solve this problem, we next formulate it mathematically.

Let there be N jobs of moving a container from the vessel to the stack. Here we define Φ to be the set of jobs with cardinality of $|\Phi| = N$. In the hybrid flow shop problem, the process of each operation in each stage has a time relationship: for a machine to process a job in a certain stage, there is a time constraint for the preceding job and the successive job. For a certain job processed in different stages, there also exists a time constraint to guarantee the sequence of operations in different stages. These time constraints are modeled as follows:

$$a_i + R(1-\sigma_{0i}^1) \geq 0 \qquad \forall i \in \Phi \qquad (4.1)$$

$$a_j + R(1-\sigma_{ij}^1) \geq b_i \qquad \forall i \in \Phi, \forall j \in \Phi, i \neq j \qquad (4.2)$$

$$a_i + t_i^{11} + t_i^{12} \leq b_i \qquad \forall i \in \Phi \qquad (4.3)$$

$$b_j + R(1-\sigma_{ij}^2) \geq c_i + t_i^{22} \qquad \forall i \in \Phi, \forall j \in \Phi, i \neq j \qquad (4.4)$$

$$b_i + t_i^{21} \leq c_i \qquad \forall i \in \Phi \qquad (4.5)$$

$$c_j + R(1-\sigma_{ij}^3) \geq c_i + t_i^{31} + t_i^{32} \qquad \forall i \in \Phi, \forall j \in \Phi, i \neq j, \qquad (4.6)$$

Table 4.1: The information of operation $O_i^{h_1h_2}$ in three stages. ($h_1 \in \{1,2,3\}$, $h_2 \in \{1,2\}$)

Operation	Equipment	Starting time	Ending time	Processing time
O_i^{11}	QC	a_i	$a_i + t_i^{11}$	t_i^{11}
O_i^{12}	QC	$a_i + t_i^{11}$	$a_i + t_i^{11} + t_i^{12}$	t_i^{12}
O_i^{21}	AGV	b_i	$b_i + t_i^{21}$	t_i^{21}
O_i^{22}	AGV	c_i	$c_i + t_i^{22}$	t_i^{22}
O_i^{31}	ASC	c_i	$c_i + t_i^{31}$	t_i^{31}
O_i^{32}	ASC	$c_i + t_i^{31}$	$c_i + t_i^{31} + t_i^{32}$	t_i^{32}

where, for $\forall\, i \in \Phi_1$ and $\forall\, j \in \Phi$ $(i \neq j)$,

- $\sigma_{ij}^1 = 1$ means that job j is handled directly after job i in Stage 1, otherwise $\sigma_{ij}^1 = 0$;
- $\sigma_{ij}^2 = 1$ means that job j is handled directly after job i in Stage 2, otherwise $\sigma_{ij}^2 = 0$;
- $\sigma_{ij}^3 = 1$ means that job j is handled directly after job i in Stage 3, otherwise $\sigma_{ij}^3 = 0$;
- a_i is the starting time of job i in Stage 1, i.e., the time at which the QC handling job i leaves P_i^2;
- b_i is the starting time of job i in Stage 2, i.e., the time at which the AGV handling job i leaves P_i^2;
- c_i is the starting time of job i in Stage 3, i.e., the time at which the ASC handling job i leaves P_i^3;
- $t_i^{h_1h_2}$ is the processing time of operation $O_i^{h_1h_2}$ with $h_1 \in \{1,2,3\}$, $h_2 \in \{1,2\}$;
- R is a large positive number.

where the unit of a_i, b_i, c_i and $t_i^{h_1h_2}$ is second. More details of the operations in the three stages are given in Table 4.1.

Inequality (4.1) initializes the first job processed by the QC. Inequality (4.2) describes the relation among job i and j handled by the QC. Inequality (4.3) guarantees job i is handled by an AGV after the QC. Inequality (4.5) guarantees job i is handled by an ASC after the AGV. Inequality (4.4) and (4.6) represent the relation of job i and j handled by an AGV and an ASC, respectively.

Besides the constraints on time (4.1)-(4.6), the discrete control variables σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3 also have equality constraints to guarantee that there is exactly one preceding job and one succeeding job for the same machine in each stage. However, for the first job j ($j \in \Phi$) to be processed, σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3 $(i \in \Phi, j \in \Phi, i \neq j)$ must be 0, and for the last job i ($i \in \Phi$) to be processed, σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3 $(i \in \Phi, j \in \Phi, i \neq j)$ must be 0. Therefore, we define two dummy jobs, indexed by 0 and $N+1$ [14]. Using Φ, we can define $\Phi_1 = \Phi \cup \{0\}$ and $\Phi_2 = \Phi \cup \{N+1\}$ which are used below to satisfy the additional constraints on the first job

and the last job. These constraints are formulated as follows:

$$\sum_{j\in\Phi_2} \sigma^1_{ij} = 1, \qquad \forall i \in \Phi \tag{4.7}$$

$$\sum_{i\in\Phi_1} \sigma^1_{ij} = 1, \qquad \forall j \in \Phi \tag{4.8}$$

$$\sum_{j\in\Phi} \sigma^1_{0j} = n_{\text{qc}} \tag{4.9}$$

$$\sum_{i\in\Phi} \sigma^1_{i(N+1)} = n_{\text{qc}} \tag{4.10}$$

$$\sum_{j\in\Phi_2} \sigma^2_{ij} = 1, \qquad \forall i \in \Phi \tag{4.11}$$

$$\sum_{i\in\Phi_1} \sigma^2_{ij} = 1, \qquad \forall j \in \Phi \tag{4.12}$$

$$\sum_{j\in\Phi} \sigma^2_{0j} = n_{\text{agv}} \tag{4.13}$$

$$\sum_{i\in\Phi} \sigma^2_{i(N+1)} = n_{\text{agv}} \tag{4.14}$$

$$\sum_{j\in\Phi_2} \sigma^3_{ij} = 1, \qquad \forall i \in \Phi \tag{4.15}$$

$$\sum_{i\in\Phi_1} \sigma^3_{ij} = 1, \qquad \forall j \in \Phi \tag{4.16}$$

$$\sum_{j\in\Phi} \sigma^3_{0j} = n_{\text{asc}} \tag{4.17}$$

$$\sum_{i\in\Phi} \sigma^3_{i(N+1)} = n_{\text{asc}} \tag{4.18}$$

Equality (4.7) and (4.8) represent that for each job $i \in \Phi$, there is exactly one preceding job and one succeeding job assigned to the QC. Equality (4.9) and (4.10) guarantee that exactly n_{qc} QC is employed (here $n_{\text{qc}} = 1$). Equality (4.11) and (4.12) represent that for each job $i \in \Phi$, there are exactly one preceding job and one succeeding job assigned to a particular AGV. Equality (4.13) and (4.14) guarantee that there are exactly n_{agv} AGVs in use. Equality (4.15) and (4.16) represent that for each job $i \in \Phi$, there are exactly one preceding job and one succeeding job assigned to a particular ASC. Equality (4.17) and (4.18) guarantee that there are exactly n_{asc} ASCs in use.

Using the inequalities and equalities constraints, the discrete-event dynamics of three types of equipment for unloading containers are modeled as a three-stage hybrid flow shop problem (4.1)-(4.18). In this hybrid flow shop problem, typically the completion time of job i at each point and the sequence of jobs that are processed by each piece of equipment in each stage are decisions variables. These decision variables are determined by the supervisory controller in the higher level.

Note that in the model presented above, we consider a vessel that needs to be unloaded in the case of one QC, multiple AGVs and multiple ASCs. A similar model can be applied to the operations of loading after unloading by decoupling these phases. When the unloading operation completes, i.e., the last unloaded container has been transported to the stack, the operation of loading typically starts. The similar model can be applied to the operation of

loading by changing the order "QC-AGV-ASC" into "ASC-AGV-QC" and updating t_i^{22} into t_{ij}^{22} in (4.4) .

4.2.3 lower-level continuous-time dynamics

At the higher level, the discrete-event dynamics of the interactions of the pieces of equipment for transporting containers are considered using (4.1)-(4.18). At the lower level the continuous-time dynamics of the individual pieces of equipment are considered. In this chapter, for the sake of simplicity, we assume that the dynamics of the pieces of equipment are identical with one dimensional moves (Chapter 6 will consider higher dimensional dynamics). Let the continuous-time dynamics of one piece of equipment be described as follows:

$$\dot{\mathbf{r}}(t) = g(\mathbf{r}(t), u(t)), \tag{4.19}$$

where $\mathbf{r}(t)$ is the continuous state, $u(t)$ is the control variable of the piece of equipment and g is a function. More specifically, let the dynamics of the piece of equipment be given by:

$$\dot{r}_1(t) = r_2(t), \tag{4.20}$$

$$\dot{r}_2(t) = u(t), \quad r_2(t) \in [v_{\min}, v_{\max}], \quad u(t) \in [u_{\min}, u_{\max}], \tag{4.21}$$

where $r_1(t)$ [m] and $r_2(t)$ [m/s] describe the position and the velocity of the piece of equipment, respectively, $u(t)$ [m/s^2] represents the acceleration, $[v_{\min}, v_{\max}]$ is the constraint on $r_2(t)$, and $[u_{\min}, u_{\max}]$ is the constraint on the acceleration.

4.3 Hierarchical Controller

Based on the decomposed dynamics, the container terminal control architecture can be described in terms of two levels in a hierarchical structure (see Fig. 4.4):

- The higher level

  The higher level controller consists of the supervisory controller and the stage controller for each stage. The supervisory controller schedules time windows of processing operations in each stage by means of determining the sequence of jobs. The stage controller assigns the time window of each operation to a particular piece of equipment. In order to schedule the operations, the supervisory controller needs to know how much time performing each operation to be scheduled requires. The higher level therefore requests from the stage controller controlling the dynamics of the equipment at the lower level the minimal time required for performing a particular operation. Based on this required time received from the stage controller and the discrete-event dynamics, the supervisory controller schedules the operations. The resulting schedule minimizes the makespan. The schedule provides the time window at which each operation is processed by each stage. Based on the time window to process all operations in each stage, the time window of an operation assigned to a particular piece of equipment is decided upon by the stage controller. The time window for performing one operation by each piece of equipment is then sent by the stage controller to the controller at the lower level.

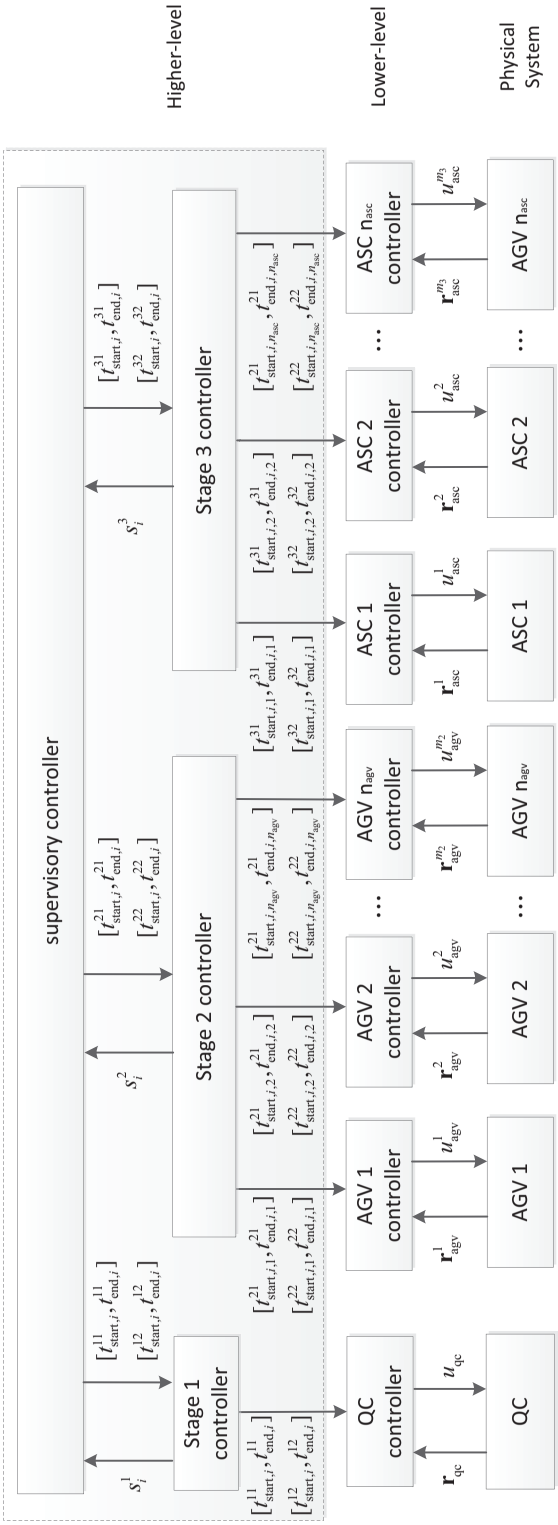


Figure 4.4: The hierarchical control architecture.

- The lower level

  At the lower level, the system is driven by the continuous-time dynamics of each piece of equipment. After receiving the time window for performing a particular operation from the higher level, the lower-level controller of a piece of equipment regards the received time window as the time constraint to perform the operation. Based on the continuous-time dynamics and a certain cost function defined over the received time window, optimal control is then proposed to be applied for the continuous-time control of the equipment.

The details of the controllers at the higher level and at the lower level are given next.

4.3.1 The higher-level controller

The higher-level controller aims at achieving an energy-efficient scheduling by maximizing the processing time of all operations when the minimal makespan is obtained. The maximized processing time of a particular operation results in energy reduction of equipment when the operation is performed. The higher-level controller contains the supervisory controller responsible for the minimal makespan and the stage controller to assign each operation to a particular piece of equipment in each stage.

Supervisory controller

The goal of the supervisory controller is to achieve the minimal makespan in an energy-efficient way. On the one hand, the objective is to minimize the makespan referring to the completion time of all n jobs. In the scheduling problem considered in this chapter, the makespan is defined as the maximal value of the completion time of all jobs in Stage 2 and the completion time of all jobs in Stage 3. In other words, it is defined as max $\{c_1+t_1^{22}, c_1+t_1^{31}+t_1^{32}, ..., c_N+t_N^{22}, c_N+t_N^{31}+t_N^{32}\}$, i.e., $\|\mathbf{w}\|_\infty$ where
$\mathbf{w} = \left[c_1+t_1^{22}, c_2+t_2^{22}, \ldots, c_N+t_N^{22}, c_1+t_1^{31}+t_1^{32}, c_2+t_2^{31}+t_2^{32}, \ldots, c_N+t_N^{31}+t_N^{32}\right]^{\mathrm{T}}$
and $\|\cdot\|_\infty$ denotes the infinity norm.

On the other hand, the objective is to minimize energy consumption. Therefore the goal of the supervisory controller at the higher level is to achieve these two objectives subject to the discrete-event dynamics, as described in Section 2.2. The time window of operations that performed by each stage and the sequence of jobs that are process by each stage are determined by the supervisory controller. To illustrate the difference between the traditional schedule and the proposed energy-efficient schedule, these two schedules will be described respectively as follows:

- The traditional scheduling problem

  The goal of the traditional scheduling problem we consider is to minimize the makespan (the completion time of all jobs) subject to the discrete-event dynamics. In such a scheduling problem the processing time of each operation is fixed (see, e.g., [14]). After defining

  $\mathbf{a} = \left[a_1, a_2, \cdots, a_N\right]^{\mathrm{T}}$, $\mathbf{b} = \left[b_1, b_2, \cdots, b_N\right]^{\mathrm{T}}$, $\mathbf{c} = \left[c_1, c_2, \cdots, c_N\right]^{\mathrm{T}}$,
  σ: the vector of $\{\sigma_{ij}^1, \sigma_{ij}^2, \sigma_{ij}^3\}_{i\in\Phi_1, j\in\Phi_2, i\neq j}$,

the traditional scheduling problem can be written as follows:

$$\min_{\mathbf{a},\mathbf{b},\mathbf{c},\sigma} \|\mathbf{w}\|_\infty \tag{4.22}$$

subject to

$$(4.1)-(4.6) \text{ and } (4.7)-(4.18).$$

- The proposed energy-efficient scheduling

  Instead of the fixed value typically considered, the processing time of each operation can be varying. Due to the interaction of different types of equipment, one type of equipment may need to wait until another type of equipment is available. For a two-point transport with both the initial speed and the final speed zero the maximal speed can be reduced when the operation time is longer. This results in the reduction of kinetic energy, when the processing time increases while the waiting time decreases. Still, considering the conflict between processing time and energy consumption, the processing time of an operation by one piece of equipment depends on the whole schedule of all pieces of equipment. In this way, the processing time of an operation in each stage (i.e., $t_i^{h_1h_2}$ $h_1 \in \{1,2,3\}, h_2 \in \{1,2\}$) can be more flexible while keeping the same makespan. Therefore, the objective to maximize the sum of the processing times of each operation subject to the minimal makespan is achieved. Considering operations O_i^{11} and O_i^{12} are identical to be processed by the same piece of equipment, for simplicity we let $t_i^{11} = t_i^{12}$. Similarly, we let $t_i^{31} = t_i^{32}$ for operations O_i^{31} and O_i^{32}.

  Here we define the sum of all operation time as $\|\mathbf{t}\|_1$, where

  $\mathbf{t} = \left[t_1^{11},\cdots,t_N^{11},t_1^{12},\cdots,t_N^{12}t_1^{21},\cdots,t_N^{21},t_1^{22},\cdots,t_N^{22},t_1^{31},\cdots,t_N^{31},t_1^{32},\cdots,t_N^{32}\right]^{\mathrm{T}}$

  and $\|\cdot\|_1$ is the 1-norm.

  Subject to the discrete-event dynamics, this optimization problem can be rewritten as follows:

$$\max_{\mathbf{t},\mathbf{a},\mathbf{b},\mathbf{c},\sigma} \|\mathbf{t}\|_1 \tag{4.23}$$

  subject to

$$\min_{\mathbf{t},\mathbf{a},\mathbf{b},\mathbf{c},\sigma} \|\mathbf{w}\|_\infty \tag{4.24}$$

$$s_i^{h_1h_2} \le t_i^{h_1h_2}, \forall i \in \Phi \tag{4.25}$$

$$t_i^{11} = t_i^{12}, \forall i \in \Phi, \tag{4.26}$$

$$t_i^{31} = t_i^{32}, \forall i \in \Phi, \tag{4.27}$$

  and subject to

$$(4.1)-(4.6) \text{ and } (4.7)-(4.18),$$

  where $s_i^{h_1h_2}$ $(h_1 = 1,2,3, h_2 = 1,2)$ is the lower bound of $t_i^{h_1h_2}$. The lower bound $s_i^{h_1h_2}$ is obtained by the stage controller by means of calculating the minimal operation

time, as detailed in Section 4.3.2. Objective (4.23) represents that the processing time of all operations should be maximized.

To solve the bi-level optimization problem (4.23)-(4.31), firstly we solve the traditional scheduling problem (6.24) subject to its constraints to obtain the minimal makespan. The first step is to relax the constraint of the minimal makespan such that the obtained minimal makespan can be located in the constraint as the equality constraint (28). The minimal makespan is obtained by solving a classical operation research problem by fixing the processing time. In other words, the integer variables of the bi-level optimization problem will be determined in the first step. It is formulated as follows:

$$\min_{\mathbf{a},\mathbf{b},\mathbf{c},\sigma} \|\mathbf{w}\|_\infty \tag{4.28}$$

subject to

$$s_i^{h_1h_2} = t_i^{h_1h_2}, \forall i \in \Phi \tag{4.29}$$

$$t_i^{11} = t_i^{12}, \forall i \in \Phi, \tag{4.30}$$

$$t_i^{31} = t_i^{32}, \forall i \in \Phi, \tag{4.31}$$

and subject to

$$(4.1)-(4.6) \text{ and } (4.7)-(4.18),$$

Here we define the result of the minimization problem above as $w_{\min}$. So we can set $\|\mathbf{w}\|_\infty = w_{\min}$ in the constraint (28) of the original optimization problem. In the meantime, σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3 obtained in the first step will be used as inputs of the optimization problem in the second step. In the second step the maximization of the processing time of operations under the minimal makespan is formulated as follows:

$$\max_{\mathbf{t},\mathbf{a},\mathbf{b},\mathbf{c}} \|\mathbf{t}\|_1 \tag{4.32}$$

subject to

$$\|\mathbf{w}\|_\infty = w_{\min} \tag{4.33}$$

$$s_i^{h_1h_2} \le t_i^{h_1h_2}, \forall i \in \Phi \tag{4.34}$$

$$t_i^{11} = t_i^{12}, \forall i \in \Phi \tag{4.35}$$

$$t_i^{31} = t_i^{32}, \forall i \in \Phi \tag{4.36}$$

and subject to

$$(4.1)-(4.6) \text{ and } (4.7)-(4.18),$$

Obtaining the operation time $t_i^{h_1h_2}$ accompanied with the starting time a_i, b_i and c_i. The time window of operation $\mathrm{O}_i^{h_1h_2}$ in three stages can be described as in Table 4.1. Here we introduce $t_{\text{start},i}^{h_1h_2}$ and $t_{\text{end},i}^{h_1h_2}$ ($h_1 \in \{1,2,3\}$, $h_2 \in \{1,2\}$) as the starting time and the ending time to process the operation $\mathrm{O}_i^{h_1h_2}$. Therefore, based on the starting time and the ending time to process an operation in three stages as shown in Table 4.1, These time windows will be then sent to the stage controller of three stages to assign each job to the specific equipment in each stage.

Stage controller

The function of the stage controller is to collect the minimal time to process each operation in the stage and to assign the time window to the specific equipment to perform operations in each stage. Since there is one QC, the time windows to perform two operations in Stage 1 are still $[t^{11}_{\text{start},i}, t^{11}_{\text{end},i}]$ and $[t^{12}_{\text{start},i}, t^{12}_{\text{end},i}]$.

Considering all AGVs are identical in Stage 2, there is no difference between the choice of AGVs. We define $\Psi_{\text{agv}} = \{1, 2, ..., n_{\text{agv}}\}$ representing the set of AGVs.

$$f_{\text{agv}} : \Phi \to \Psi_{\text{agv}}, \tag{4.37}$$

where f_{agv} is a function that maps the set of jobs Φ to the set of AGVs Ψ_{agv}. $f_{\text{agv}}(i)$ describes the particular AGV assigned to job i. The time windows of job i assigned to a particular AGV can be denoted by $[t^{21}_{\text{start},i,f_{\text{agv}}(i)}, t^{21}_{\text{end},i,f_{\text{agv}}(i)}]$ and $[t^{22}_{\text{start},i,f_{\text{agv}}(i)}, t^{22}_{\text{end},i,f_{\text{agv}}(i)}]$.

Regarding the ASCs, the mapping of job i to the machine is predetermined since each container has a certain origin in the vessel and a certain destination in the stack. Here we use $f_{\text{asc}}(i)$ for describing the assigned ASC for job i. Then the time windows of job i assigned to a particular ASC is denoted by $[t^{31}_{\text{start},i,f_{\text{asc}}(i)}, t^{31}_{\text{end},i,f_{\text{asc}}(i)}]$ and $[t^{32}_{\text{start},i,f_{\text{asc}}(i)}, t^{32}_{\text{end},i,f_{\text{asc}}(i)}]$..

4.3.2 The lower-level controllers

At the lower level, each piece of equipment has a controller that decides on the continuous-time trajectory of a piece of equipment. In each controller at the lower level, an optimal control problem is formulated so as to complete the operation given by the higher level within the operation time allowed. The controller at the lower level can hereby take into account additional objectives, such as energy consumption minimization. The specific control problem of a piece of equipment depends on the operation that it has to carry out. In particular, in this case depending on the task and the position from where to start to where to go, the moving behaviors of each piece of equipment are different.

Let r_0 and r_{f} denote the origin and destination positions of a piece of equipment. The initial state and the final state are denoted as $\mathbf{r_0} = \begin{bmatrix} r_0 & 0 \end{bmatrix}^{\text{T}}$ and $\mathbf{r}_{\text{f}} = \begin{bmatrix} r_{\text{f}} & 0 \end{bmatrix}^{\text{T}}$. For an operation processed by one piece of equipment, the corresponding piece of equipment will process the operation from $\mathbf{r_0}$ to $\mathbf{r}_{\text{f}}$ within the given time window. This is an optimal control problem that can be formulated as follows:

$$\min_{u(t)} J(\mathbf{r}(t), u(t)) \tag{4.38}$$

subject to

$$\dot{\mathbf{r}}(t) = g(\mathbf{r}(t), u(t)), \tag{4.39}$$

$$\mathbf{r}(t_0) = \mathbf{r}_0, \quad \mathbf{r}(t_f) = \mathbf{r}_f, \quad t \in [t_0, t_f], \tag{4.40}$$

where $J(\mathbf{r}(t), u(t)) = \int_{t_0}^{t_{\text{f}}} 0.5 m r_2(t)^2$ is the objective function quantifying the energy consumption with mass m and velocity r_2. The piece of equipment starts its operation at t_0 and has to complete its operation before t_{f}. The initial and the final state in (4.40) guarantee the operation is completed. Here, (4.39) represents the continuous-time dynamics of the piece of equipment as explained in Section 4.2.2.

Solving the control problem

To solve the optimal control problem above (4.38), the numerical approach is chosen. In the numerical approach, this continuous optimal control problem can be simplified to an optimization problem and the optimal solution can be obtained by available solvers efficiently. Considering the system model is a linear system, the energy consumption problem in this chapter can be formulated as a standard quadratic programing problem. Such quadratic programming problems can be solved efficiently.

The objective of this optimal control problem is to minimize the mechanical energy of the piece of equipment from the origin state r_0 to its final state t_f over the time horizon $[t_0, t_\mathrm{f}]$ for operation $\mathrm{O}_i^{h_1 h_2}$. For discretization, the time step size is defined as ΔT and then $\frac{t_i^{h_1 h_2}}{\Delta T} + 1$ is the number of discretized steps over $[t_0, t_\mathrm{f}]$. The discretized dynamical model, based on (4.20) and (4.21), for a piece of equipment is as follows: for time instant k

$$\mathbf{r}(k+1) = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \mathbf{r}(k) + \begin{bmatrix} 0.5\Delta T^2 \\ \Delta T \end{bmatrix} u(k) = \mathbf{A}\mathbf{r}(k) + \mathbf{B}u(k), \tag{4.41}$$

where $\mathbf{r}(k) = \begin{bmatrix} r_1(k) & r_2(k) \end{bmatrix}^\mathrm{T}$ describes the position $r_1(k)$ and velocity $r_2(k)$ of the piece of equipment, and $u(k)$ is the acceleration of the piece of equipment, $\mathbf{A}$ and $\mathbf{B}$ are referred to system matrix and input matrix, respectively.

This optimization problem is to minimize the mechanical energy of the piece of equipment from $k = 0$ to $k = N_\mathrm{s}$ subject to its dynamics and constraints. After defining $\mathbf{u} = [u(0), u(1), \ldots, u(N_\mathrm{s} - 1)]^\mathrm{T}$, the optimization problem can be formulated as follows:

$$\min_{\mathbf{u}} \sum_{k=1}^{N_\mathrm{s}} 0.5m(r_2(k))^2, \tag{4.42}$$

subject to, for $k = 0, 1, ..., N_\mathrm{s} - 1$,

$$\mathbf{r}(k+1) = \mathbf{A}\mathbf{r}(k) + \mathbf{B}u(k), \tag{4.43}$$

$$\mathbf{r}_\mathrm{min} \leq \mathbf{r}(k) \leq \mathbf{r}_\mathrm{max}, \tag{4.44}$$

$$u_\mathrm{min} \leq u(k) \leq u_\mathrm{max}, \tag{4.45}$$

$$\mathbf{r}(0) = \mathbf{r}_0, \quad \mathbf{r}(N_\mathrm{s}) = \mathbf{r}_\mathrm{f}. \tag{4.46}$$

where $0.5m(r_2(k))^2$ describes the kinetic energy of the piece of equipment at time k, $\mathbf{r}_\mathrm{min}$ and $\mathbf{r}_\mathrm{max}$ are the constraints on states $\mathbf{r}(k)$ and u_min and u_max are the constraints on the control variable $u(k)$.

The optimization problem above is a quadratic programing problem. When this problem is solved, the lower level controller will set the calculated trajectories as the reference for the piece of equipment.

Determining the minimal-time required

The hierarchical control architecture proposed provides a methodology for achieving the minimal makespan in an energy-efficient way. In this control architecture, the minimal time of each operation is required at the higher level for scheduling operations. As the interaction between the higher level and the lower level of the hierarchical control architecture,

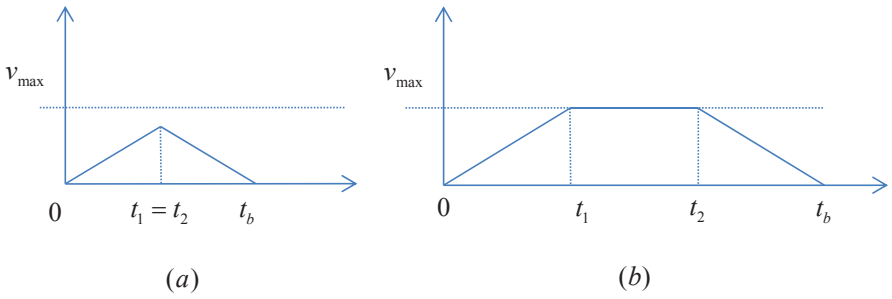


Figure 4.5: The minimal-time profile for a different distance d_t. (a): $d_t < \frac{v_{max}^2}{u_{max}}$, $(b) : d_t \geq \frac{v_{max}^2}{u_{max}}$.

the lower bound of the processing time of each operation by one piece of equipment needs to be computed. The minimal time required for processing an operation by one piece of equipment depends on the states and continuous-time dynamics of the piece of equipment. The minimal-time required to complete a certain operation can be obtained from the theory of optimal control, as the result of Pontryagin's Minimum Principle. Application of Pontryagin's Minimum Principle results in the minimization of two so-called Hamiltonian functions. Details of the theory behind this principle can be found in, e.g., [35]. Here we provide the outcome of applying this principle in our setting. In our setting, application of the principle yields the control action $u(t)$ that minimizes the time for carrying out a task as follows:

$$u(t) = \begin{cases} -u_{max} & \text{for } t = t_2^+, \ldots, t_b \\ 0 & \text{for } t = t_1^+, \ldots, t_2^- \\ u_{max} & \text{for } t = 0^+, \ldots, t_1^-, \end{cases} \tag{4.47}$$

where t_1 and t_2 are so-called switching points between different control modes, $t_1^+ \geq t_1 + \varepsilon$, $t_2^+ \geq t_2 + \varepsilon$, $t_1^- \geq t_1 - \varepsilon$ and $t_2^- \geq t_2 - \varepsilon$ (ε is a small positive value), and where t_1 in and t_b are calculated as:

$$t_1 = \begin{cases} \frac{v_{max}}{u_{max}} & \text{if } d_t \geq \frac{v_{max}^2}{u_{max}} \\ \sqrt{\frac{d_t}{u_{max}}} & \text{if } d_t < \frac{v_{max}^2}{u_{max}} \end{cases} \tag{4.48}$$

$$t_b = \begin{cases} 2\frac{v_{max}}{u_{max}} + \frac{d_t - \frac{v_{max}^2}{u_{max}}}{v_{max}} & \text{if } d_t \geq \frac{v_{max}^2}{u_{max}} \\ 2\sqrt{\frac{d_t}{u_{max}}} & \text{if } d_t < \frac{v_{max}^2}{u_{max}}. \end{cases} \tag{4.49}$$

The different minimal-time profiles for carrying out the task with respect to distance d_t can be seen in Fig. 4.5. The minimal-time depends on the relationship between d_t and $\frac{v_{max}^2}{u_{max}}$ (which is obtained based on the optimal control action). t_b is the ending time and the distance d_t is used as the integration over $[0, t_b]$.

In summary, the minimal-time required to process an operation by one piece of equipment can be obtained by Pontryagin's necessary conditions. This minimal-time required is

given by t_b above. It is the lower bound on the time required for processing an operation by a piece of equipment. This bound is sent to the higher level as $s_i^{h_1 h_2}$ ($h_1 = 1,2,3, h_2 = 1,2$) for scheduling the operations.

4.3.3 Control architecture summary

Summarizing, the control problem for three types of equipment is decomposed into two levels with the following steps. First once the stage controller receives the request from supervisory controller for the time required for processing an operation, the lower bound of the operation time is determined by the stage controller by solving the minimal-time control problem. Later the supervisory controller determines the processing time of operations in each stage. The scheduling problem is formulated by the supervisory controller as a three-stage hybrid flow shop problem. Then the time windows for processing operations in each stage are sent from the supervisory controller to the stage controller. The stage controller assigns each operation to a particular piece of equipment in each stage. The pieces of equipment subsequently carry out the operation, possibly also taking into account energy saving objectives. The complete procedure of the hierarchical control structure includes the following steps, as illustrated in Fig. 4.6.

Step 1 The supervisory controller requests the time required for processing an operation in each stage;

Step 2 The minimal time for processing an operation by the QC, the AGV and the ASC is calculated by the stage controller;

Step 3 The supervisory controller receives the minimal time from the stage controller;

Step 4 The supervisory controller computes the energy-efficient schedule for all jobs in three stages;

Step 5 The supervisory controller sends the processing time of operations to the stage controller;

Step 6 The stage controller assigns each operation to a particular piece of equipment in each stage;

Step 7 The lower-level controller of QCs, AGVs and ASCs receive the time window to compute the trajectories of each piece of equipment for energy saving and the whole procedure completes.

4.3.4 Heuristic control of equipment

The previous section proposed the hierarchical control architecture to achieve the minimal makespan in an energy efficient way. The hierarchical architecture emphasizes the interdependence of the scheduling problem regarding the discrete-event dynamics of all pieces of equipment and the optimal control problem with respect to the individual piece of equipment considering the continuous-time dynamics.

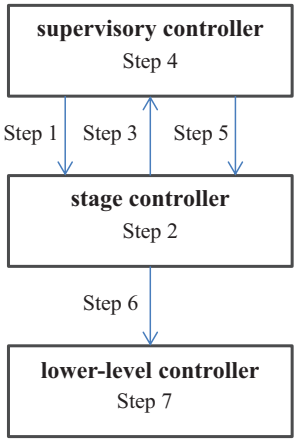


Figure 4.6: The flow diagram of all these steps for the hierarchical control architecture.

Earlier works [14, 17] focus on the schedule with regards to the discrete-event dynamics, wherein the scheduling and the control of the individual piece of equipment are considered separately and the continuous-time dynamics of the individual piece of equipment is typically simplified. For instance, Cao et al. [14] use the constant travel speed to compute the travel time of equipment. Such a simplification could result in difficulties of implementing the simplified control of equipment when the dynamics and constraints of equipment are considered (e.g., the speed and the acceleration). In particular, it could lose the improvement of energy-efficiency which the acceleration and the speed account for.

Since the operational control of container terminals in the literature focuses on the discrete-event dynamics for scheduling, we include the scheduling of all pieces of equipment and the optimal control of the continuous-time dynamics of the individual pieces of equipment in order to make a consistent comparison with the proposed hierarchical control. The optimal control of the individual piece of equipment is considered as the minimal-time control mentioned in Section 4.3.2 since the scheduling is typically considered to minimize the completion time of all jobs [14, 17] .

For the scheduling the mathematical problem formulation for the scheduling of all piece of equipment without considering energy saving is presented in Section 4.3.1, referred to as the traditional scheduling. Besides the mathematical approach, some heuristic approaches used in practice for scheduling are also considered for comparison. Considering the position of the containers in the vessel, here we consider the closest selection and the random selection in Stage 1 as the heuristic approaches in comparison with the energy-efficient schedule:

- Closest container selection

  The sequence of N jobs σ_{ij}^1 in Stage 1 ranges from the closest place to the furtherest place in the container vessel.

- Random container selection

  The sequence of N jobs σ_{ij}^2 in Stage 1 is determined randomly.

The heuristic approaches above for scheduling could be integrated with the minimal-time control of the individual piece of equipment for comparing the performance of different approaches consistently. In the following part, both the closest approach and the random approach for the operational control of equipment contain the minimal time of operations in each stage.

In the integrated heuristic methods, we use the same hierarchical architecture as we summarized in Section 3.3. The differences of the integrated heuristic methods lie in Step 4 and Step 7. In Step 4, the supervisory controller will use the heuristic approach to determine the schedule for all jobs in three stages. In Step 7, the processing time for each piece of equipment can not be changed as in the proposed energy-efficient approach.

4.4 Simulation experiments

In this section, different approaches for the operational control of the medium-size container terminal will be tested and evaluated. In particular, the performance of the traditional schedule and the energy-efficient schedule discussed in Section 4.3.1 will be compared. For the medium-size container terminal, Benchmark System 2 proposed in Chapter 2 is chosen in this chapter. The completion time, the energy consumption and the utilization of the QC, the AGVs and the ASCs will be used as KPIs in this chapter for evaluating the performance.

4.4.1 Setup

In Benchmark System 2 (see Section 2.4), one QC, two AGVs and three ASCs are used for unloading a particular bay of a container vessel. (two AGVs are enough to support high utilization of the QC). The settings of this benchmark system are presented as follows:

- The initial position of the QC is set to its unloading position. The initial position of the AGV is set to its loading position. The initial position of the ASC is set to its loading position;
- Eight containers are considered to be transported with the same arrival time;
- The processing time of one container by the QC depends on the specific position in the vessel;
- Each container here considered has the same vertical position in the vessel;
- The storage location of each container to be transported is generated randomly;
- The containers stored in each stack have different storage places;
- The service time of the QC, the AGV and the ASC for exchanging containers are ignored.

Based on the setup of the simulation, ten experiments are carried out for an overall conclusion for the performance comparison of different approaches.

The scheduling problem at the higher-level controller is solved by the solver CPLEX in the OPTI toolbox [21]. The quadratic programing problem at the lower-level controller is solved by the solver OOQP in the OPTI Toolbox [21].

Table 4.2: The main results of performance indicators generated by the traditional schedule using the hierarchical controller architecture.

	The traditional schedule		
	completion time (s)	energy consumption (kWh)	computation time (s)
Test 1	477	8.16	37
Test 2	476	10.15	157
Test 3	496	11.61	123
Test 4	478	10.17	852
Test 5	476	10.46	31
Test 6	481	11.31	126
Test 7	467	9.40	162
Test 8	503	10.53	107
Test 9	462	8.94	92
Test 10	468	10.05	32

4.4.2 Results and discussion

The general results of simulations, in terms of performance indicators, are given in Table 4.2 and Table 4.3. Table 4.6 compares the proposed methodology with some heuristic approaches in the minimal makespan. The comparison of energy consumption is shown in Fig. 4.7. As the illustration to compare the schedule, the results of two approaches chosen from Test 6 are presented in Fig. 4.8 and Fig. 4.9.

Table 4.2 and Table 4.3 compare the main performance of the traditional schedule and the energy-efficient schedule using the hierarchical controller architecture. Using the energy-efficient schedule , the average energy consumption is reduced by 37% for the same minimal completion time. The piece of equipment aims to slow down to reduce energy consumption. Therefore, the waiting time of the piece of equipment is decreased as much as possible and results in utilization improvement of all pieces of equipment, as presented in Table 4.5. Note that the computation time in Table 4.3 includes the second computation after the first computation for the minimal completion time, which is given in Table 4.2.

Table 4.6 shows the differences in the completion time between the proposed methodology and some heuristic approaches. Among this comparison in Table 4.6, the proposed methodology is superior to the closest and the random approach with respect to the completion time.

Fig. 4.7 presents the energy consumption of the traditional schedule and the energy-efficient schedule both by the hierarchical controller architecture. In general, the energy consumption is reduced remarkably by the proposed energy-efficient schedule using the hierarchical controller architecture. In the traditional schedule, the processing time of one operation by each piece of equipment is fixed in which the piece of equipment is operated at its maximal velocity subject to its maximal acceleration. This approach results in significant energy consumption, compared to the energy-efficient schedule. In the energy-efficient schedule, each piece of equipment does not need to work at its maximal speed. Instead, the piece of equipment has more flexible processing time without loss of the minimal completion time. In such a flexible processing time, the piece of equipment can reduce the mechanical energy required to transport a specific container. In particular, the amount of

Table 4.3: The main results of performance indicators generated by the energy-efficient schedule using the hierarchical controller architecture.

	The energy-efficient schedule		
	completion time (s)	energy consumption (kWh)	computation time (s)
Test 1	477	4.82	37+0.15
Test 2	476	7.05	157+0.06
Test 3	496	8.82	123+0.06
Test 4	478	5.15	852+0.06
Test 5	476	5.13	31+0.07
Test 6	481	6.54	126+0.06
Test 7	467	4.97	162+0.06
Test 8	503	7.61	107+0.06
Test 9	462	6.76	92+0.06
Test 10	468	6.29	32+0.06

Table 4.4: The equipment utilization of the traditional schedule.

	The traditional schedule		
	QC utilization	AGV utilization	ASC utilization
Test 1	100%	58%	68%
Test 2	100%	57%	70%
Test 3	100%	56%	81%
Test 4	100%	58%	46%
Test 5	100%	58%	52%
Test 6	100%	57%	61%
Test 7	100%	54%	53%
Test 8	92%	51%	73%
Test 9	100%	60%	57%
Test 10	100%	56%	67%

energy consumption of ASCs is reduced considerably due to the great amount of weight of the ASC. Because of the large energy contribution from ASCs, the total energy consumption is reduced largely by the proposed energy-efficient schedule.

Fig. 4.8 and Fig. 4.9 illustrate the comparisons between the traditional schedule and the energy-efficient scheduling both by the hierarchical controller architecture. The number in each block is associated with the number of container. In the traditional schedule, the processing time of one operation by the piece of equipment is fixed. As a result, some pieces of equipment have more waiting time due to the synchronization of different types of equipment. This can be seen in the schedule of AGVs and ASCs in Fig. 4.8. In contrast, the piece of equipment has more flexible processing time as shown in Fig. 4.9. Therefore, the time to perform an operation by one piece of equipment can increase, which is beneficial to the energy reduction. It is noticeable that the reduction of energy consumption does not result in the increase of the completion time. Instead, the completion time in both Fig. 4.8 and Fig. 4.9 are the same.

Table 4.5: The equipment utilization of the energy-efficient schedule.

	The energy-efficient schedule		
	QC utilization	AGV utilization	ASC utilization
Test 1	100%	100%	100%
Test 2	100%	100%	100%
Test 3	100%	100%	100%
Test 4	100%	100%	100%
Test 5	100%	100%	100%
Test 6	100%	100%	100%
Test 7	100%	100%	100%
Test 8	100%	100%	100%
Test 9	100%	100%	100%
Test 10	100%	100%	100%

Table 4.6: The completion time with respect to different approaches.

	optimal	closest	random
Test 1	477s	477s	496s
Test 2	476s	542s	552s
Test 3	496s	573s	540s
Test 4	478s	502s	478s
Test 5	476s	516s	504s
Test 6	481s	546s	520s
Test 7	467s	551s	488s
Test 8	503s	570s	543s
Test 9	462s	476s	532s
Test 10	468s	490s	539s

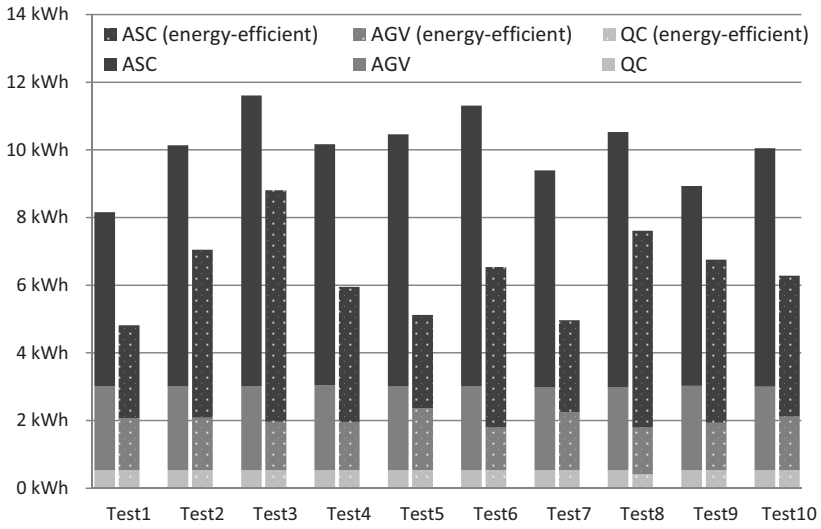


Figure 4.7: The results of energy consumption generated by the two different schedules using the hierarchical controller architecture. The left column and the right column of each test refer to energy consumption of the traditional schedule and the proposed energy-efficient schedule, respectively.

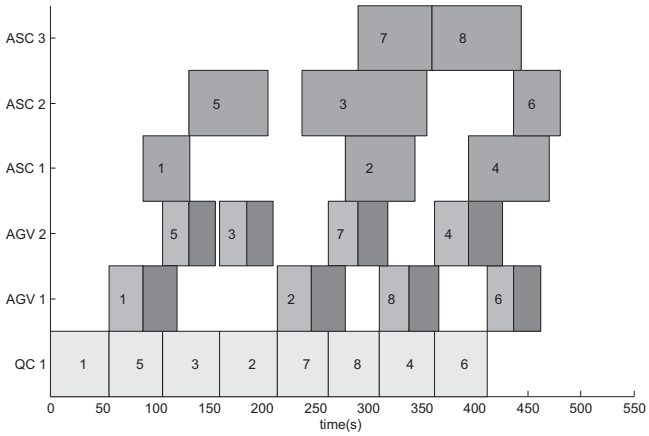


Figure 4.8: The traditional schedule as determined by the hierarchical control architecture in Test 6.

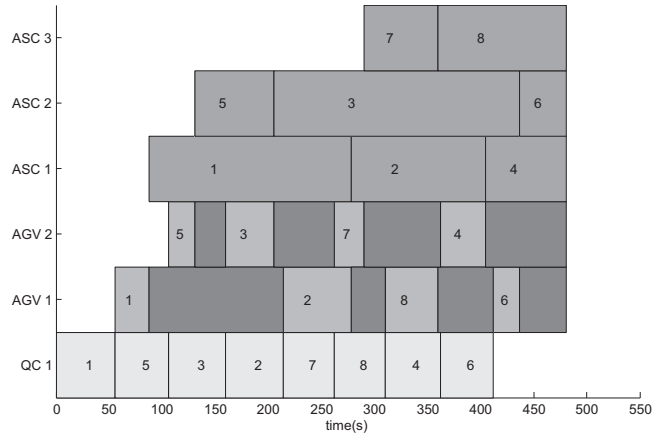


Figure 4.9: The energy-efficient schedule as determined by the hierarchical control architecture in Test 6.

4.5 Concluding remarks

In this chapter, the energy efficiency of the medium-size container terminal is considered and an open-loop control problem is investigated. The dynamics of transporting containers in the medium-size container terminal are considered as the combination of including continuous-time and discrete-event dynamics. A hierarchical control architecture is proposed, consisting of two levels. The higher level is responsible for scheduling in which the processing time of an operation by a particular piece of equipment is determined; the lower level consists of controller per piece of equipment for an optimal control problem. At the higher level, the minimal completion time is obtained; at the lower level, the energy consumption reduction is achieved if possible, while satisfying time constraints on processing an operation given by the higher level. Simulation results show that on average the proposed approach can save 37% of energy consumption for the same minimal completion time in comparison with the traditional scheduling. The reduction of energy consumption results from the flexible schedule by making use of waiting time when two types of equipment need to be synchronized.

Although the energy efficiency of the medium-size container terminal has been improved, the control problem considered in this chapter is an open-loop case which cannot handle uncertainties during real-time operations. Also, the number of containers considered in this chapter is small and in practice a large number of containers needs to processed. Chapter 5 will investigate this problem, i.e., improving energy efficiency during real-time operations of the medium terminal. The hierarchical architecture proposed in this chapter will be used as basis for the framework in Chapter 5.

In this chapter, in the case of one QC a small number of AGVs are employed and collision avoidance is not necessarily considered. However, when multiple QCs are considered a large number of AGVs are involved and collision avoidance must be considered. Chapter 6 will investigate this research problem.

Chapter 5

Event-driven model predictive control for real-time operations

Chapter 4 discussed the energy efficiency of the medium-size container terminal at the operational level in which an open-loop control problem has been investigated. This chapter continues with this system, while focusing on the closed-loop operational control. For the real-time operational control, the computation burden needs to be reduced and operational uncertainties must be taken into account. For addressing these problems, a receding horizon principle will be applied in this chapter.

The research discussed in this chapter is based on [111, 114].

5.1 Introduction

This chapter concerns real-time decisions for energy efficiency of the medium-size container terminal. In the previous chapter, energy efficiency was achieved by coordinating the control of individual pieces of equipment and the schedule of all pieces of equipment to be employed for transporting a number of containers. The investigation on energy efficiency in Chapter 4 is considered as an open-loop case without any disturbance. However, during the transport of containers, operational uncertainties can change the process of the transporting containers and influence the energy efficiency of the container handling system. The involved uncertainties can be operation delay, the precise time at which new containers arrive, the breakdown of the equipment, etc. Therefore, real-time operations must be determined to adjust changes in the dynamically operating environment of container terminals. Furthermore, solving hybrid flow shop scheduling problems, such as those in the higher-level of the hierarchical control architecture is known to be NP-hard [86]. This results in heavy computational burden when a large number of containers are involved in the bay handling of the vessel, whereas in Chapter 4 a small number of containers are considered.

To handle operational uncertainties and reduce the computational burden, an event-driven Model Predictive Control (MPC) is proposed in this chapter for scheduling and rescheduling all operations involved for transporting a number of containers in an energy efficient way. The trajectory relevant to each operation is hereby determined on-line by re-

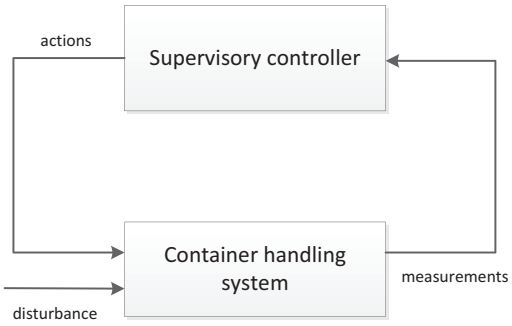


Figure 5.1: The structure of the rescheduling scheme for highlighting the supervisory controller.

ceiving the updated operations times from the supervisory controller, which is highlighted in Fig. 5.1. Two common types of uncertainties, i.e., the operational delay and the precise arrival time of new containers, are handled by this MPC controller. The proposed MPC controller reduces the computational burden significantly compared to the open-loop scheduling problem. Similarly as in Chapter 4, the completion time of transporting all containers is referred to as makespan.

The remainder of this chapter is organized as follows: Section 5.2 describes the discrete-event dynamics and continuous-time dynamics of the container terminal separately. Section 5.3 proposes an event-driven MPC controller for carrying out the operations of jobs. Section 5.4 compares the performance of the proposed MPC controller and the existing controller. Section 5.5 concludes this chapter with remarks.

5.2 Modeling of equipment

5.2.1 Modeling of interacting machines

Chapter 4 considers the case of unloading a vessel and presents the discrete-event dynamics of the medium-size container terminal are modeled as a three-stage hybrid flow shop. In that model, two successive operations are considered for a particular QC or ASC (i.e., O_i^{11} and O_i^{12} for the QC in Stage 1, O_i^{31} and O_i^{32} for the ASC in Stage 3). When real-time decisions are to be determined, on the one hand two types of equipment are required to be coordinated and on the other hand these two operations for the QC or the ASC also needs to be coordinated. We extend the discrete-event model of Chapter 4 and merge these two operations for the QC or the ASC into one single operation. Therefore, the operation of an individual machine is emphasized for rescheduling operations of interacting machines, simplifying the coordination of the discrete-event dynamics. The merged operation is related to a particular optimal control problem for one individual piece of equipment that will be discussed in Section 5.3.2 and Section 5.3.3.

As a three-stage hybrid flow shop, the operations of the three types of machines are described in terms of three stages:

1. Stage 1: one QC

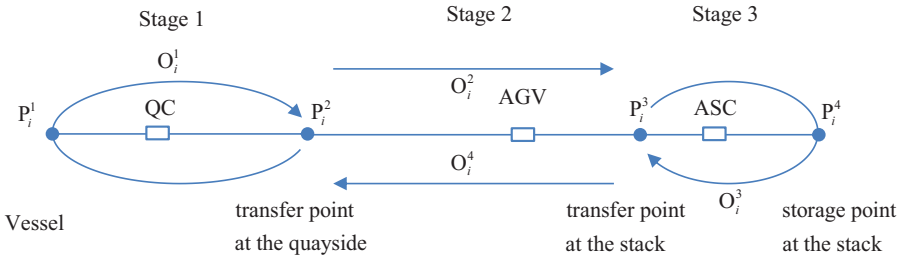


Figure 5.2: The sequence of transporting containers using three types of machines.

2. Stage 2: multiple AGVs

3. Stage 3: multiple ASCs

The operations of the three types of equipment are illustrated in Fig. 5.2. P_i^1 is defined as the place of container i in the vessel. P_i^2 is the defined as the transfer point at which container i is transferred from a QC to an AGV. P_i^3 is defined as the transfer point at which container i is transferred from an AGV to an ASC. P_i^4 is defined as the storage place of container i in the stack.

We consider one job involving four operations, as shown in Fig. 5.2. Each operation only requires a single machine operation and the transition of the two operations involves the interaction of two types of machines. In Stage 1, O_i^1 is defined as the move of the QC from P_i^2 to P_i^2 throughout P_i^1 for picking up a container in the vessel. In Stage 2, there are two operations O_i^2 and O_i^4 in which an AGV moves from P_i^2 to P_i^3 with container i and the AGV returns from P_i^3 to P_i^2 after unloading container i, respectively. Operations O_i^3 is defined in Stage 3, in which an ASC transports container i from P_i^3 to P_i^3 throughout P_i^4 for releasing the container into the storage place.

As same as Chapter 4, we still consider N jobs of moving a container from vessel to stack, Φ to be the set of jobs (cardinality $|\Phi| = N$). Two dummy jobs 0 and $N+1$ are still used for defining $\Phi_1 = \Phi \cup \{0\}$ and $\Phi_2 = \Phi \cup \{N+1\}$. Due to the extension for defining the operations, time constraints for a particular piece of equipment are described as follows:

$$a_i + R(1-\sigma_{0i}^1) \geq 0 \qquad \forall i \in \Phi \tag{5.1}$$

$$a_j + R(1-\sigma_{ij}^1) \geq b_i \qquad \forall i \in \Phi, \forall j \in \Phi \tag{5.2}$$

$$a_i + t_i^1 \leq b_i \qquad \forall i \in \Phi \tag{5.3}$$

$$b_j + R(1-\sigma_{ij}^2) \geq c_i + t_i^4 \qquad \forall i \in \Phi, \forall j \in \Phi \tag{5.4}$$

$$b_i + t_i^2 \leq c_i \qquad \forall i \in \Phi \tag{5.5}$$

$$c_j + R(1-\sigma_{ij}^3) \geq c_i + t_i^3 \qquad \forall i \in \Phi, \forall j \in \Phi, \tag{5.6}$$

where, for $\forall\, i \in \Phi_1$ and $\forall\, j \in \Phi$ $(i \neq j)$,

- $\sigma_{ij}^1 = 1$ means that job j is handled directly after job i in Stage 1, otherwise $\sigma_{ij}^1 = 0$;
- $\sigma_{ij}^2 = 1$ means that job j is handled directly after job i in Stage 2, otherwise $\sigma_{ij}^2 = 0$;

- $\sigma_{ij}^3 = 1$ means that job j is handled directly after job i in Stage 3, otherwise $\sigma_{ij}^3 = 0$;
- a_i is the starting time of job i in Stage 1, i.e., the time at which the QC handling job i leaves P_i^2;
- b_i is the starting time of job i in Stage 2, i.e., the time at which the AGV handling job i leaves P_i^2;
- c_i is the starting time of job i in Stage 3, i.e., the time at which the ASC handling job i leaves P_i^3;
- t_i^h is the processing time of operation O_i^h with $h \in \{1,2,3,4\}$;
- R is a large positive number.

where the unit of a_i, b_i, c_i and t_i^h is second.

Inequality (5.1) initializes the first job processed by the QC. Inequality (5.2) describes the relation among job i and j handled by the particular QC. Inequality (5.3) guarantees that job i is handled by an AGV after a QC. Inequality (5.5) guarantees that job i is handled by an ASC after an AGV. Inequalities (5.4) and (5.6) represent the relation of job i and j handled by a particular AGV and a particular ASC, respectively.

As same as Chapter 4, additional equality constraints of the discrete decision variables σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3 are described as (4.7) to (4.18).

Using these inequalities and equalities constraints, the discrete-event dynamics of three types of machines are modeled as a three-stage hybrid flow shop. In this hybrid flow shop, the completion time of job i processed by each stage and the sequence of jobs that are processed by each machine in each stage are decisions variables. These decision variables will be determined by the supervisory controller discussed below.

For the continuous-time dynamics of an individual machine, we still consider the double integrator in discrete time, which is the same presented as (4.41) in Chapter 4.

5.3 Receding horizon controller

This section proposes an event-driven receding horizon supervisory controller for scheduling and rescheduling the interacting machines, as represented in Fig. 5.1. The information from the handling system to the supervisory controller consists of the lower bounds of required times for carrying out the operations involved in the various jobs considered by the supervisory controller. The control actions of the supervisory controller include setting the allowed operation times of uncompleted jobs and dispatching of new jobs. Here a disturbance refers to the uncertainty due to the operation delay and the arrival of new containers.

The supervisory controller is considered part of the higher-level controller of the hierarchical architecture proposed in Chapter 4. Initially the supervisory controller provides the solution for scheduling part of containers before all operations begin. Later an event triggers the rescheduling of the supervisory controller. The supervisor will then measure the actual operation times depending on the states of the machines, i.e., the actual position and the actual velocity of the machines. These measurements are then used to update the minimal time needed for completing the ongoing jobs. Then, the supervisory controller updates

the operation times involved in completing the considered jobs. The supervisory controller subsequently determines the new schedule for the interacting machines.

5.3.1 Supervisory controller

The supervisory controller determines the energy-efficient schedule of the interacting machines by solving an optimization problem. In the optimization problem, we consider an objective function combining the minimization of the makespan and the maximization of the sum of all operation times. In this chapter, we consider the case in which the jobs are carried out exactly in the order as requested by the shipper, similarly as in [59]. The decision variables are the operation times and the job sequences in Stage 2 and Stage 3.

In the considered scheduling problem, the makespan is defined as the maximal value of the completion time of all jobs in Stage 2, carried out by the AGVs, and the completion time of all jobs in Stage 3, carried out by the ASCs. In other words, it is defined as max $\{c_1+t_1^3,...,c_N+t_N^3,c_1+t_1^4,...,c_N+t_N^4\}$, i.e., $\|\mathbf{w}\|_\infty$, where $\mathbf{w}=[c_1+t_1^3,c_2+t_2^3,\ldots,c_N+t_N^3,c_1+t_1^4,c_2+t_2^4\ldots,c_N+t_N^4]^{\mathrm{T}}$ and $\|\cdot\|_\infty$ denotes the infinity norm.

After defining

$$\mathbf{a}=\left[a_1,a_2,\cdots,a_N\right]^{\mathrm{T}}$$

$$\mathbf{b}=\left[b_1,b_2,\cdots,b_N\right]^{\mathrm{T}}$$

$$\mathbf{c}=\left[c_1,c_2,\cdots,c_N\right]^{\mathrm{T}}$$

$\boldsymbol{\sigma_1}$: the vector of $\{\sigma_{ij}^1\}_{i\in\Phi_1,j\in\Phi_2,i\neq j}$

$\boldsymbol{\sigma_2}$: the vector of $\{\sigma_{ij}^2\}_{i\in\Phi_1,j\in\Phi_2,i\neq j}$

$\boldsymbol{\sigma_3}$: the vector of $\{\sigma_{ij}^3\}_{i\in\Phi_1,j\in\Phi_2,i\neq j}$

$\mathbf{t}$: the vector of $\{t_i^h\}_{i\in\Phi,h=1,2,3,4}$

this scheduling problem can be written as follows:

$$\min_{\mathbf{a},\mathbf{b},\mathbf{c},\boldsymbol{\sigma_2},\boldsymbol{\sigma_3},\mathbf{t}} \|\mathbf{w}\|_\infty-\lambda\|\mathbf{t}\|_{\mathbf{1}} \tag{5.7}$$

subject to

$$\begin{aligned} s_i^h \le t_i^h, \text{for } h\in\{1,2,3,4\} \\ (5.1)-(5.6) \text{ and } (4.7)-(4.18) \end{aligned} \tag{5.8}$$

where s_i^h is the lower bound of t_i^h ($h\in\{1,2,3,4\}$), n is the number of jobs to be processed, λ is a small penalty on the sum of all operation times. For each operation, the operation time t_i^h and its starting time a_i, b_i or c_i constitute the time windows given in Table 5.1.

The scheduling problem formulated above is the foundation of the rescheduling approach discussed later in Section 5.3.4.

Table 5.1: The time windows of operations in three stages.

Operation	Machine	Starting time	Ending time
O_i^1	QC	a_i	$a_i + t_i^1$
O_i^2	AGV	b_i	$b_i + t_i^2$
O_i^3	ASC	c_i	$c_i + t_i^3$
O_i^4	AGV	c_i	$c_i + t_i^4$

5.3.2 Minimal-time calculation

This section discusses how the minimal time of a particular operation (O_i^1, O_i^2, O_i^3 or O_i^4) is computed, wherein the operation can be ongoing or unprocessed (i.e., the operation has not started yet). This minimal time is used for determining the lower bound of the actual operation time allowed for this particular operation.

For the minimal time calculation problem, in Chapter 4 we consider the simple case of the double integrator for individual piece of equipment, including the overall move from the origin to the destination without a stop during the move. For this case, the result of Pontryagins Minimum Principle in [35] can be directly applied for solving this problem. In this chapter, a different case in which the piece of equipment possible may have a different initial state and an optional stop is considered and we hereby propose a numerical method for calculating this minimal time.

In this minimal time calculation problem, the machine (QC, AGV or ASC) with an initial state $\mathbf{r}_0$ at time instant k_0 is required to reach the target $\mathbf{r}_f(\mathbf{r}_f = [r_{1,\mathrm{f}}, 0])$ as fast as possible. By introducing the binary variable $b_\mathrm{t}(k)$, the minimal time required by a machine to complete the operation can be obtained as follows: $\forall k \in [k_0+1, ..., k_0+T_\mathrm{w}]$,

$$\begin{aligned} &r_1(k) - r_{1,\mathrm{f}} \leq R(1-b_\mathrm{t}(k)) \\ &r_1(k) - r_{1,\mathrm{f}} \geq -R(1-b_\mathrm{t}(k)) \\ &r_2(k) - 0 \leq R(1-b_\mathrm{t}(k)) \\ &r_2(k) - 0 \geq -R(1-b_\mathrm{t}(k)) \end{aligned} \tag{5.9}$$

$$\sum_{k=k_0+1}^{k_0+T_\mathrm{w}} b_\mathrm{t}(k) = 1, \forall k \in [k_0+1, ..., k_0+T_\mathrm{w}] \tag{5.10}$$

$$\mathbf{r}_(t_0) = \mathbf{r}_0, \tag{5.11}$$

where T_w is a significantly large number which ensures the calculation of the minimal time and R is a large positive number to ensures that the constraints in (5.9) are active only when $b_\mathrm{t}(k) = 1$. Equations (5.10) and (5.11) force the state $\mathbf{r}$ of a machine to move from the initial state $\mathbf{r}_0$ to the target $\mathbf{r}_\mathrm{f}$.

If we define $e(k)$ as the elapsed time at time instant k ($e(k) = (k-k_0)\Delta T$), then $e(k)b_\mathrm{k}(k)$ describes the finishing time when $b_\mathrm{t}(k) = 1$. Therefore, the minimal time of an operation can be obtained by minimizing the sum of finishing times according to different $\mathbf{r}_f$ as follows:

$$\min_{\mathbf{u},\mathbf{b}_\mathrm{t}} \sum_{k=k_0+1}^{k_0+T_\mathrm{w}} e(k)b_\mathrm{t}(k), \tag{5.12}$$

subject to (4.41) and (5.9)–(5.11),
where $e(k)$ is the elapsed time at time instant k, $\mathbf{u}$ and $\mathbf{b}_{\mathrm{t}}$ are continuous and binary control variables of optimization problem (5.12).

We define the minimal time since t_0 for completing these four operations (O_i^1, O_i^2, O_i^3, O_i^4) as $\hat{s}_i^1$, $\hat{s}_i^2$, $\hat{s}_i^3$ and $\hat{s}_i^4$, respectively. The value of the objective function in (5.12) gives the minimal time $\hat{s}_i^2$ and $\hat{s}_i^4$ for completing O_i^2 and O_i^4 directly. For operation O_i^1 and O_i^3, they are both a particular two-point transport problem. We define $t_{i,\min}^1$ and $t_{i,\min}^3$ as the minimal time for completing the whole operation O_i^1 and O_i^3, and $s_{i,\mathrm{cal}}^1$ and $s_{i,\mathrm{cal}}^3$ as the result of (5.12) for O_i^1 and O_i^3. The detailed $\hat{s}_i^1$ and $\hat{s}_i^3$ for completing O_i^1 and O_i^3 are described as follows:

$$\hat{s}_i^1 = \begin{cases} s_{i,\mathrm{cal}}^1 + 0.5t_{i,\min}^1 & \text{before the middle point } \mathrm{P}_i^1 \\ s_{i,\mathrm{cal}}^1 & \text{after the middle point } \mathrm{P}_i^1 \end{cases} \tag{5.13}$$

$$\hat{s}_i^3 = \begin{cases} s_{i,\mathrm{cal}}^3 + 0.5t_{i,\min}^3 & \text{before the middle point } \mathrm{P}_i^4 \\ s_{i,\mathrm{cal}}^3 & \text{after the middle point } \mathrm{P}_i^4 \end{cases} \tag{5.14}$$

The values of k_0, $\mathbf{r}_0$ and $\mathbf{r}_{\mathrm{f}}$ and k_0 depend on if the operation is ongoing or unprocessed. If the operation is ongoing, k_0 is the time instant triggered for rescheduling, $\mathbf{r}_0$ is measured as the state of the machine and $\mathbf{r}_{\mathrm{f}}$ is given depending on the specific operation. If the operation is unprocessed, these values are exactly the same as planned off-line.

5.3.3 Energy-efficient optimal control

This section presents the mathematical formulation of the optimal control problem for energy reduction. In general, within the remained time steps T_{r} for completing the operation determined by the supervisory controller, an optimal control problem is formulated based on its initial state $\mathbf{r}_0$ at time t_0 and the target $\mathbf{r}_{\mathrm{f}}$ at time instant $k_0 + T_{\mathrm{r}}$ of the particular operation.

This optimal control problem is also applicable to these four operations (O_i^1, O_i^2, O_i^3 and O_i^4) when they are either ongoing or unprocessed. k_0 and $\mathbf{r}_0$ rely on whether the operation is ongoing or unprocessed, which is the same as considered in the minimal time problem above. $\mathbf{r}_{\mathrm{f}}$ are related to one of the transfer points P_i^2 and P_i^3. Due to the differences between the AGV operation and the crane operation, the energy minimization problems are discussed separately in the following parts.

Energy-efficient optimal control for AGV

For AGVs, operations O_i^2 and O_i^4 involve two-point transport problems. For a double integrator dynamical system, the energy minimization problem is to minimize the sum of absolute values of the accelerations [35], while reaching the destination. Therefore, the minimal energy optimal control problem for AGVs can be formulated as follows:

$$\min_{\mathbf{u}} \sum_{k=k_0}^{k_0+T_{\mathrm{r}}-1} \|u(k)\| \tag{5.15}$$

subject to

$$(4.41),$$
$$\mathbf{r}(k_0) = \mathbf{r}_0, \tag{5.16}$$
$$\mathbf{r}(k_0 + T_\mathrm{r}) = \mathbf{r}_\mathrm{f}, \tag{5.17}$$

where $\mathbf{u}$ are continuous control variables of the optimization problem. Equations (6.14) and (6.15) force the machine to move from the initial state $\mathbf{r}_0$ to the target $\mathbf{r}_f$.

This optimization problem is a linear programming problem, the result of which is easily obtained using existing solvers.

Energy-efficient optimal control for cranes

For the QC and the ASC, O_i^1 and O_i^3 both involve a special two-point transport problem, in which a particular crane moves from one point to the second point and then back to the first point. Therefore, we consider a particular optimization formulation differing from the formulation O_i^2 and O_i^4.

In this minimal energy control problem, a crane (QC or ASC) is required to move from its initial state $\mathbf{r}_0$ at time instant k_0 to the target $\mathbf{r}_\mathrm{f}$ (throughout the middle point P_i^1 or P_i^4 if applicable for a stop) using less energy within a given time window. We define $\mathbf{r}_m(\mathbf{r}_m = [r_{1,\mathrm{m}}, 0])$ as the stopping state of the middle point. The stop of $\mathbf{r}_m$ is formulated using binary variable $b_\mathrm{c}(t)$ as follows: $\forall k \in [k_0+1, ..., k_0+T_\mathrm{r}]$,

$$\begin{aligned} &r_1(k) - r_{1,\mathrm{m}} \le R(1 - b_\mathrm{c}(k)) \\ &r_1(k) - r_{1,\mathrm{m}} \ge -R(1 - b_\mathrm{c}(k)) \\ &r_2(k) - 0 \le R(1 - b_\mathrm{c}(k)) \\ &r_2(k) - 0 \ge -R(1 - b_\mathrm{c}(k)) \end{aligned} \tag{5.18}$$

$\forall k \in [k_0+1, ..., k_0+T_\mathrm{r}-1]$,

$$\sum_{k=k_0+1}^{k_0+T_\mathrm{r}-1} b_\mathrm{c}(k) = \begin{cases} 1 & \text{before the middle point } \mathrm{P}_i^1 \text{ or } \mathrm{P}_i^4 \\ 0 & \text{after the middle point } \mathrm{P}_i^1 \text{ or } \mathrm{P}_i^4 \end{cases} \tag{5.19}$$

$$\mathbf{r}(k_0) = \mathbf{r}_0 \tag{5.20}$$
$$\mathbf{r}(k_0 + T_\mathrm{r}) = \mathbf{r}_\mathrm{f}, \tag{5.21}$$

where R is the large and positive number to guarantee the constraints in (5.18) are active only when $b_\mathrm{c}(k) = 1$. Equations (5.20) and (5.21) force the crane to move from the initial state $\mathbf{r}_0$ to the target $\mathbf{r}_\mathrm{f}$.

Therefore, this optional minimal-energy optimal control problem can be formulated as follows:

$$\min_{\mathbf{u},\mathbf{b}_\mathrm{c}} \sum_{k=k_0}^{k_0+T_\mathrm{r}-1} \|u(k)\| \tag{5.22}$$

subject to

$$(4.41), (5.18) - -(5.21),$$

Algorithm 1 Event-driven energy-efficient algorithm

```
Initialize the first N_p jobs (k_d = 0)
while |Φ(k_d)| = N_p do
    if t = ||d_1||_∞ then
        k_d = k_d + 1
        Remove conditional completed jobs
        Add new unprocessed jobs (|Φ(k_d)| ≤ N_p)
        Compute s_i^h locally for event k_d (i ∈ Φ(k_d))
        Update s_i^h for event k_d (i ∈ Φ(k_d))
        Reschedule and determine t_i^h for event k_d
        Execute t_i^h locally for event k_d
    end if
end while
```

where $\mathbf{u}$ and $\mathbf{b}_c$ are continuous and binary control variables of the optimization problem, respectively. This optimization problem is a mixed integer linear programming problem, which can be solved by efficient solvers such as CPLEX.

5.3.4 Event-triggered rescheduling algorithm

The supervisory controller concerns the discrete-event dynamics of the container handling system. For a class of discrete-event systems, [98] proposed the event-driven MPC strategy. Similarly, in this chapter we propose a receding horizon strategy for energy-efficient rescheduling based on triggered events. Here the event is defined as the moment at which the operations of the first job processed by the QC are completed. Using a receding horizon principle, we consider the supervisory controller responses for a fixed set of containers $|\Phi(k_d)| = N_p$ for event k_d. The procedures of scheduling can be found in Algorithm 1.

The algorithm involves several key steps:

1. In the initialization ($k_d = 0$), the schedule of the first N_p jobs is made.
2. At event k_d, the supervisory controller excludes the job which is first processed by the QC and of which all operations are complete. This continues until the first job processed by the QC does not completes all its operations.
3. The supervisory controller adds new unprocessed jobs to $\Phi(k_d)$.
4. The local controller of the machine computes the minimal time for completing the operation $\hat{s}_i^h$ at time t and update s_i^h for the whole operation as in Table. 5.2.
5. The updated s_i^h is sent to the supervisory controller as the lower-bound for processing these operations.
6. The supervisory controller determines the updated operation times and sends them back to the local controller of the machines.
7. The local controller of each machine adjusts its acceleration considering the updated operation times, as given in Table. 5.3.

Table 5.2: The parameters of the minimal time problem for the operations.

operation	ongoing	unprocessed
O_i^1	$k_0 = t/\Delta T$, $s_i^1 = \hat{s}_i^1 + t - a_i$	$k_0 = a_i/\Delta T$, $s_i^1 = \hat{s}_i^1$
O_i^2	$k_0 = t/\Delta T$, $s_i^2 = \hat{s}_i^2 + t - b_i$	$k_0 = b_i/\Delta T$, $s_i^2 = \hat{s}_i^2$
O_i^3	$k_0 = t/\Delta T$, $s_i^3 = \hat{s}_i^3 + t - c_i$	$k_0 = c_i/\Delta T$, $s_i^3 = \hat{s}_i^3$
O_i^4	$k_0 = t/\Delta T$, $s_i^4 = \hat{s}_i^4 + t - c_i$	$k_0 = c_i/\Delta T$, $s_i^4 = \hat{s}_i^4$

Table 5.3: The parameters of the energy-efficient problem for the operations.

operation	ongoing	unprocessed
O_i^1	$k_0 = t/\Delta T$, $T_\mathrm{r} = (t_i^1 - t + a_i)/\Delta T$	$k_0 = a_i$, $T_\mathrm{r} = t_i^1/\Delta T$
O_i^2	$k_0 = t/\Delta T$, $T_\mathrm{r} = (t_i^2 - t + b_i)/\Delta T$	$k_0 = b_i$, $T_\mathrm{r} = t_i^2/\Delta T$
O_i^3	$k_0 = t/\Delta T$, $T_\mathrm{r} = (t_i^3 - t + c_i)/\Delta T$	$k_0 = c_i$, $T_\mathrm{r} = t_i^3/\Delta T$
O_i^4	$k_0 = t/\Delta T$, $T_\mathrm{r} = (t_i^4 - t + c_i)/\Delta T$	$k_0 = c_i$, $T_\mathrm{r} = t_i^4/\Delta T$

This scheduling algorithm does not update the planning anymore when the number of containers to be planned is no more than the prediction horizon N_p as few containers are left.

5.3.5 Blocking control

In the literature, rescheduling for operations of container terminals is hardly investigated. In [11] a rescheduling policy has been proposed for large-scale task allocation of autonomous straddle carriers, in which the rescheduling does not change the schedule of planned jobs. However, the rescheduling of straddle carriers is different from the case of QC-AGV-ASC for which there is no particular rescheduling method. As for comparison we consider a basic approach for dynamic scheduling in general manufacturing systems, referred to as the periodic policy [78]. This policy is regarded as a simple and effective approach since it decomposes the large scale scheduling problem into many small static scheduling problems that can be solved efficiently. The schedule is then executed and not changed until the next period starts. Considering that the scheduling of container terminals depends on the number of containers, in this chapter we decompose the scheduling of all containers into several blocks in each of which the number of containers is equal. In the following part of this chapter, this control method, referred to as blocking control (BC) is considered for comparison with the proposed MPC controller.

5.3.6 Performance indicators

Similarly as in the previous chapters, the completion time and the energy consumption (proposed in Chapter 2) are chosen as KPIs in this chapter to evaluate the effect of the proposed MPC controller.

The completion time can be calculated simply as the completion time of all containers. (See Section 5.3.1)

The energy consumption of all machines for the complete simulation is calculated as

Table 5.4: The weight parameters of the controlled components.

$m^{\mathrm{e}}_{\mathrm{qc}}$	$m^{\mathrm{f}}_{\mathrm{qc}}$	$m^{\mathrm{e}}_{\mathrm{agv}}$	$m^{\mathrm{f}}_{\mathrm{agv}}$	$m^{\mathrm{e}}_{\mathrm{asc}}$	$m^{\mathrm{f}}_{\mathrm{asc}}$
10 ton	25 ton	15 ton	30 ton	240 ton	255 ton

follows:

$$E_{\mathrm{tot}} = E^{\mathrm{sim}}_{\mathrm{qc}} + E^{\mathrm{sim}}_{\mathrm{agv}} + E^{\mathrm{sim}}_{\mathrm{asc}}, \tag{5.23}$$

where E_{tot} describes the energy consumption of all machines, $E^{\mathrm{sim}}_{\mathrm{qc}}$, $E^{\mathrm{sim}}_{\mathrm{agv}}$ and $E^{\mathrm{sim}}_{\mathrm{asc}}$ denote the energy consumption of the QC, the AGVs and the ASCs for the complete simulation, respectively. The calculation of $E^{\mathrm{sim}}_{\mathrm{qc}}$ is presented here:

$$E^{\mathrm{sim}}_{\mathrm{qc}} = \sum_{k=0}^{N_{\mathrm{s}}-1} E_{\mathrm{qc}}(k) \tag{5.24}$$

$$E_{\mathrm{qc}}(k) = \begin{cases} 0.5 m_{\mathrm{qc}}(k) \times (v^2_{\mathrm{qc}}(k+1) - v^2_{\mathrm{qc}}(k)) & v^2_{\mathrm{qc}}(k+1) > v^2_{\mathrm{qc}}(k) \\ 0 & \text{else} \end{cases} \tag{5.25}$$

$$m_{\mathrm{qc}}(k) = \begin{cases} m^{\mathrm{e}}_{\mathrm{qc}} & v_{\mathrm{qc}}(k+1) < 0 \\ m^{\mathrm{f}}_{\mathrm{qc}} & v_{\mathrm{qc}}(k+1) > 0, \end{cases} \tag{5.26}$$

where N_{s} is the total simulation step, $m^{\mathrm{e}}_{\mathrm{qc}}$ is the unloading weight of the QC without the container and the $m^{\mathrm{e}}_{\mathrm{qc}}$ is the loading weight of the QC with the container. Similarly, $m^{\mathrm{e}}_{\mathrm{agv}}$ and $m^{\mathrm{e}}_{\mathrm{asc}}$ are defined as the unloading weight of the AGV and the ASC; $m^{\mathrm{f}}_{\mathrm{agv}}$ and $m^{\mathrm{f}}_{\mathrm{asc}}$ are defined as the loading weight of the AGV and the ASC. The weights of the machines [41, 42, 54] are given in Table 5.4. E_{agv} and E_{asc} can be computed in a similar way as E_{qc}.

5.4 Simulation experiments

In this section, different approaches for real-time operation control of the medium-size container will be evaluated in a number of simulations. The medium-size container terminal, Benchmark System 2 (see Section 2.4 of Chapter 2) will be used in this chapter. As the KPIs proposed in Chapter 2, the completion time and the energy consumption will be used in this chapter.

5.4.1 Setup

Next we present simulations in which the proposed MPC controller controls the system. We consider the case of 1 QC, 2 AGVs and 3 ASCs of Benchmark System 2. This configuration aims to achieve a high handling capacity, which is consistent with the setting of Chapter 4. One ASC is installed per stack. Besides the feature of Benchmark System 2 proposed in Chapter 2, additional features with respect to this chapter are as follows:

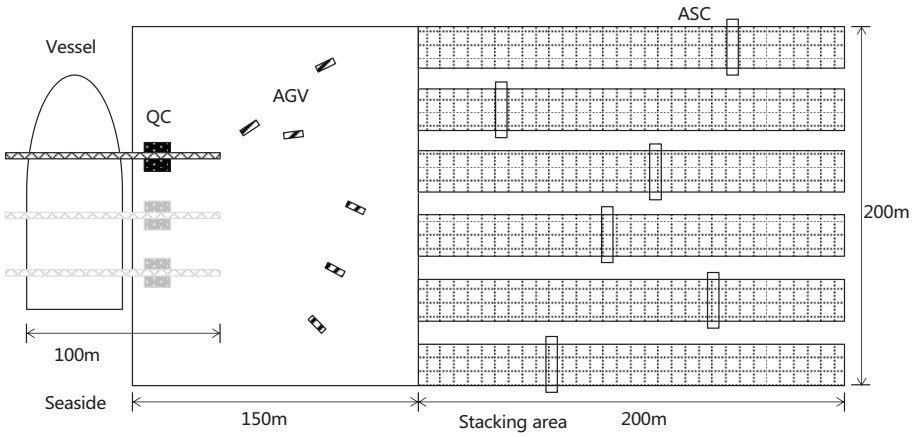


Figure 5.3: The layout of Benchmark System 2.

Table 5.5: The dynamical set of jobs and times of the scheduling over events.

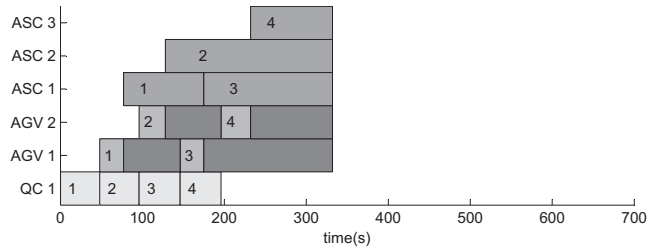
Event k_d	$\Phi(k_\mathrm{d})$	time
0	{1,2,3,4}	0s
1	{2,3,4,5}	175s
2	{3,4,5,6}	280s
3	{4,5,6,7}	363s
4	{5,6,7,8}	453s

- The traveling distance of the AGV d_agv for three stacks are 100m, 120m and 140m, respectively. This assumption has taken the suggested layout from a terminal simulation company into account;
- The handling starts when the vessel just berthed;
- The maximal computation time of the open-loop controller is set to 1 hour;
- The penalty λ in the objective function (5.22) is chosen to 0.01;
- The sample time ΔT is set to 1 second.

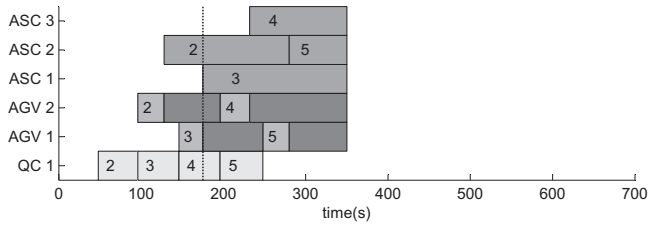
The simulations are carried out in Matlab using an Intel Core 2430 processor (2.4GHz) with 4GB memory. The involved mixed integer linear programming problem is solved by CPLEX 12.5.

5.4.2 Illustration of MPC controller

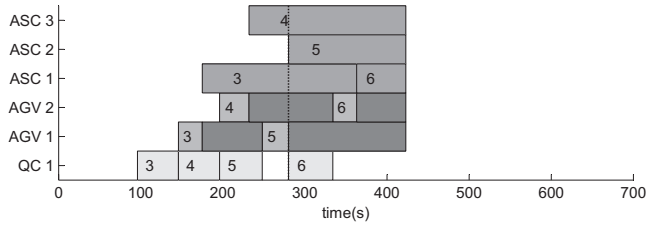
This section provides an example of the MPC controller for illustrating the receding horizon control principle. For sake of simplicity, in this example we consider the case of transporting 8 containers with N_p set to 4.



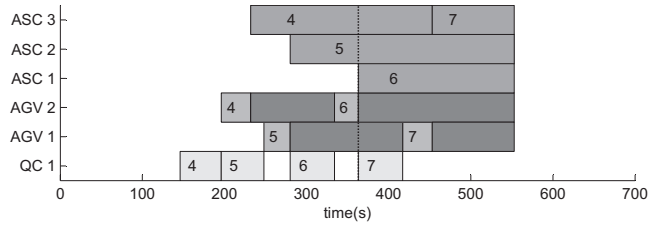
(a) The schedule of the MPC controller at Event 0.



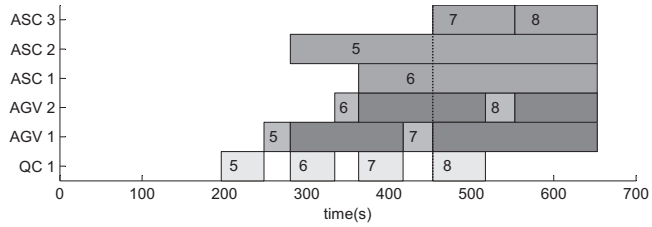
(b) The schedule of the MPC controller at Event 1.



(c) The schedule of the MPC controller at Event 2.



(d) The schedule of the MPC controller at Event 3.



(e) The schedule of MPC controller at Event 4.

Figure 5.4: The schedule of the MPC controller over events (the dashed line indicates the event moment).

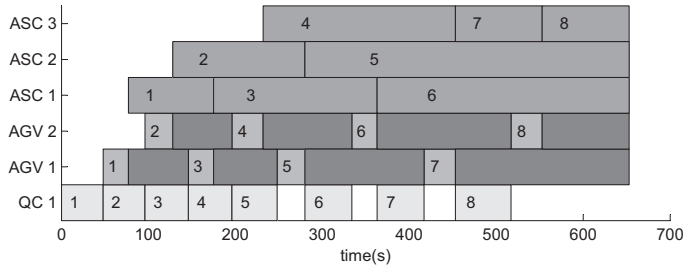


Figure 5.5: The overall result using the MPC controller.

Fig. 5.4 shows how the event-driven MPC controller plans the schedule over events and Fig. 5.5 provides the overall planning using the MPC controller. Table 5.5 presents the dynamical set of jobs considered in the MPC controller. At Event 0 (0s), the first N_p containers are planned. At Event 1 when the operations for the first job of $\Phi(0)$ have completed (175s), this job is excluded from $\Phi(k_\mathrm{d})$ and a new job is added to $\Phi(k_\mathrm{d})$, since $N_\mathrm{p} = 4$. The operation times of uncompleted operations are updated using the MPC controller. Similarly, the remaining jobs are planned in a receding horizon way.

5.4.3 Prediction horizon choice

A motivation to use the MPC control is that it can reduce the computation burden in comparison to the open-loop perspective taking into account all containers at once as in Chapter 4. In this chapter, the proposed event-based MPC controller concerns the hybrid flow shop scheduling problem which is NP-hard, even in the case when one stage contains two machines and the other stage has a single machine [86]. However, in the MPC controller N_p directly influences the performance of the controlled system and computation time which is crucial for real-time control. In the following part, we discuss the choice of N_p.

To investigate the influence of N_p on performance, we test the performance of the completion time and total energy consumption by changing N_p of the proposed MPC controller. To test different N_p, we consider the number of containers assumed to be 40 in this simulation. Due to the stochastic setting of the sequenced containers and their destinations in each stack, we conduct 5 different settings analyze the average performances for a particular N_p of the proposed MPC controller. For the five settings, we also compare performance of the MPC controller with the open-loop optimal control. The open-loop optimal control solves the entire planning once. For the five cases, the optimality of the open-loop optimal control is not achieved within the maximal computation time 1 hour.

Fig. 5.6 presents the average completion time and the average energy consumption when different N_p are considered. It is observed from Fig. 5.6 that a longer prediction horizon can result in a smaller completion time. The completion time decreases correspondingly as N_p increases when N_p is small. This is because a longer prediction horizon can take more information of consecutive containers into account and lead to a shorter completion time of transporting all containers. The completion time stabilizes when N_p is 10 since the prediction horizon is long enough for predicting necessary containers to be transported. The open-loop controller has a smaller completion time than the MPC controller, however it

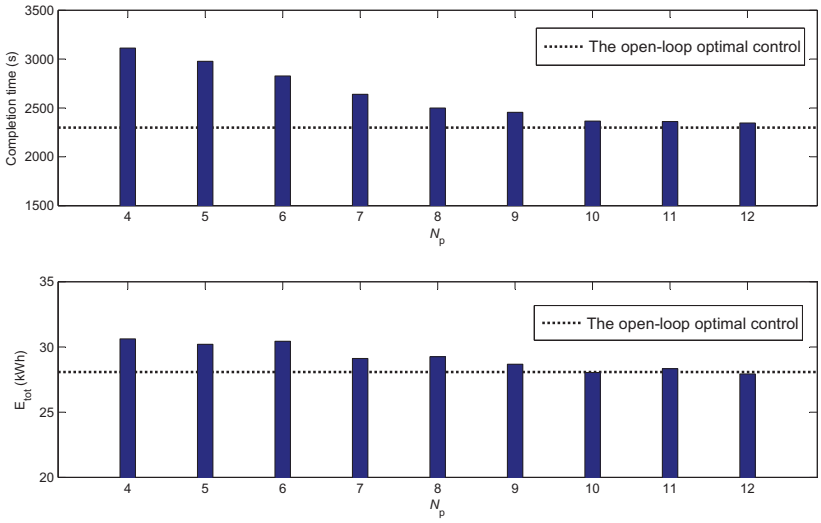
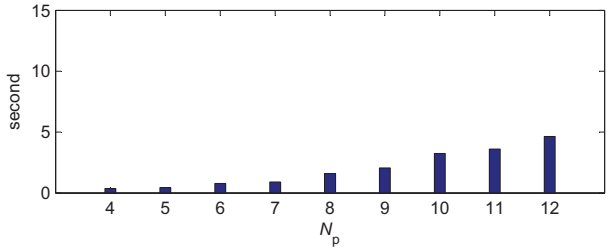
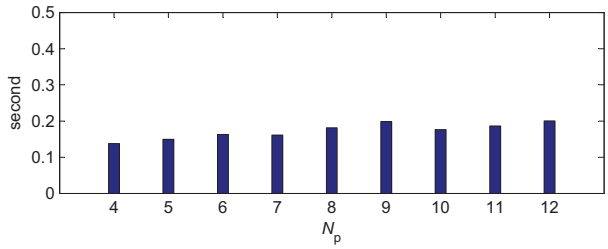


Figure 5.6: The simulation results of the total energy consumption and the completion time for varying N_p.



(a) The initial computation time of the MPC controller.



(b) The average computation time of the MPC controller for rescheduling.

Figure 5.7: The simulation results of the computation time for varying N_p.

results in the expensive computation. For energy consumption, although it fluctuates when N_p increases, the trend of declining can be observed.

Fig. 5.7 shows the average initial computation time of the MPC controller (for event 0) and the average computation time of reschedulings for varying N_p. For the initial horizon,

Table 5.6: The parameters of the ASC and the AGV for the bay handling.

	d_{asc}			d_{agv}		
Containers	stack 1	stack 2	stack 3	stack 1	stack 2	stack 3
50	56m	136m	132m	100m	120m	140m
60	189m	99m	98m			
70	136m	80m	74m			
80	35m	7m	113m			
90	109m	145m	105m			
100	33m	159m	63m			

the computation time grows significantly as the prediction horizon increases. The average computation time remains at a low number, although it rises slightly as the prediction horizon increases.

In the proposed MPC controller we choose $N_{\text{p}} = 10$ as this value balances the size of the optimization problem and computation time, while still providing a competitive performance in comparison to other prediction horizons.

5.4.4 Adaptiveness

In Section 5.4.2 we explored the effect of different N_{p} on the performance of the MPC controller for transporting a particular number of containers. Still, in practice the number of containers can vary according to the size of the bay.

Here we consider the cases of transporting containers typically considered for a bay of a container vessel. In each case, the order of containers processed by the QC and their destinations for the same stack are generated randomly. Considering that the typical order of magnitude for the number of containers for a bay is 100 as shown in [90], we test the transport of 50, 60, 70, 80, 90 and 100 containers to illustrate that the MPC controller can be applied to a generic setting of container handling for a bay of the container vessel. The related parameters can be found in Table 5.6. Parameters d_{agv} and d_{asc} are the AGV and the ASC for each stack per case.

First, Fig. 5.8 compares the completion time of the proposed MPC controller and the BC controller for receding horizon rescheduling. The proposed MPC controller can take into account the interaction of jobs between different blocks of containers and therefore it has 11% less completion time than the BC controller on average. This advantage can be found for all cases of containers to be transported.

Fig. 5.8 presents the energy consumption of the proposed MPC controller and the BC controller for receding horizon rescheduling. The MPC controller considers the block interaction of containers and as a result it also reduces the energy consumption for transporting containers than the MPC controller. In general, more container transportation results in more energy consumption. However, the storage location of the containers for each stack is generated randomly. Then the ASC does not operate at a high speed for a very short distance. This explains why the energy consumption of transporting 80 containers is significantly less than the other cases in Fig. 5.8.

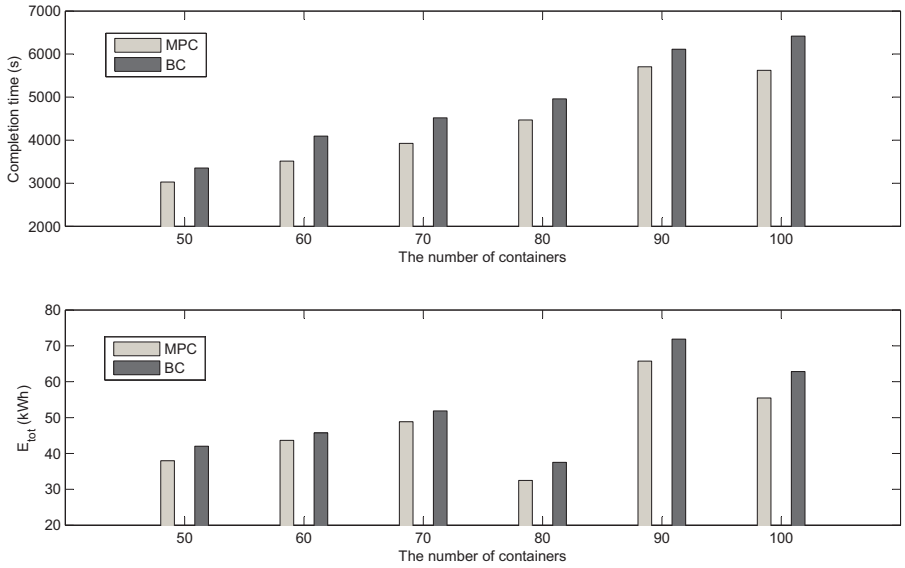


Figure 5.8: The performance comparison of transporting various containers

5.4.5 Scenario with uncertainties

Section 5.4.2 and 5.4.3 show the advantage of reducing the computation burden using the proposed MPC controller. In the simulated scenarios, all containers are transported as planned without any uncertainties. Besides the computational benefit, the MPC controller can also handle uncertainties that may take place in the container handling system. As two main common types of uncertainties, the operational delay and the precise arrival time of new containers will be investigated next. The performance of the MPC controller will be compared with the open-loop controller. In the open-loop controller, the operation times are determined beforehand and the starting time of the operation may change in case of uncertainties.

Operational delay

In the handling process of a container terminal, one machine may have a delay resulting from the handling operation (e.g., slow down the speed). This requests that the MPC controller can adapt the delayed condition and transport the remaining containers still in an energy-efficient way. In this chapter, we simulate an operation delay of 50s occurring when the ASC is processing O_2^3 in the stack due to the slower speed unexpectedly. Due to the possible delay resulting from the slower speed, the trajectories of the machines have to be re-optimized for the sake of energy-efficient handling. On the one hand the completion time needs to be minimized, while the energy reduction is expected to remain low on the other hand. To test the performance of the MPC controller, we set up a scenario in which a group of containers are transported from the vessel. In this scenario, the number of containers is assumed to be 16 and a 50-second delay takes place when O_2^3 is processed by the ASC. This scenario is simulated both for the MPC controller and the open-loop controller.

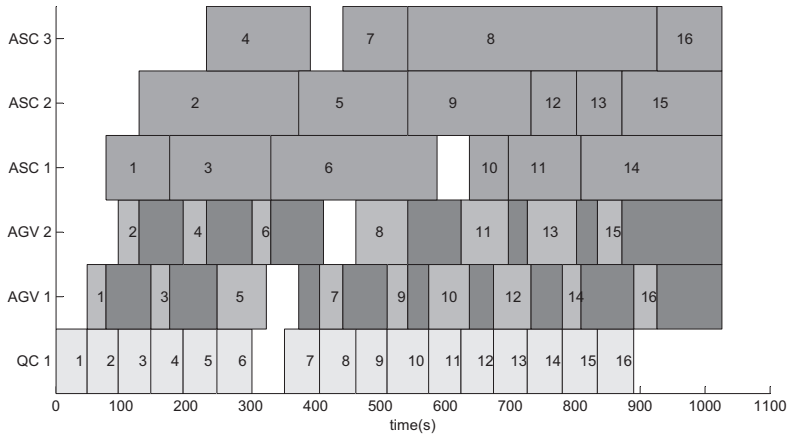


Figure 5.9: The scheduled result of the open-loop controller in the scenario of the operation delay.

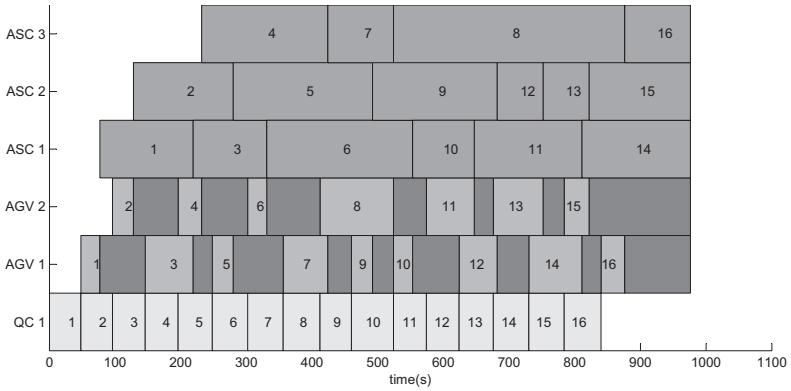


Figure 5.10: The scheduled result of the MPC controller in the scenario of the operation delay.

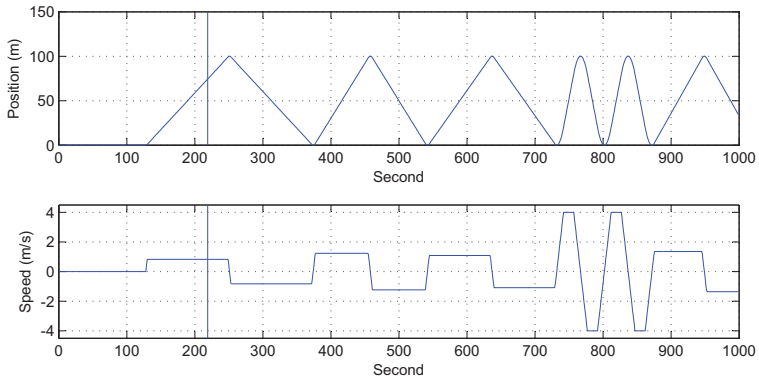


Figure 5.11: The trajectory of ASC 2 related to O_2^3 for the open-loop controller.

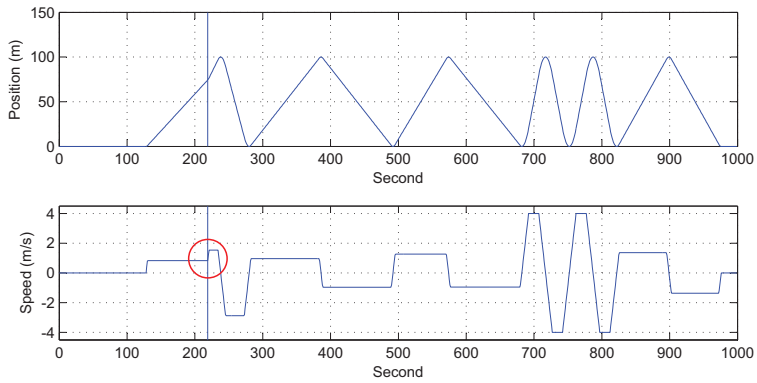


Figure 5.12: The trajectory of ASC 2 related to O_2^3 for the MPC controller.

Table 5.7: The performance of controllers in the scenario of the operation delay.

controller	MPC	Open-loop
Completion time	976s	1026s
E_{tot}	8.84kWh	9.58kWh

Fig. 5.9 and Fig. 5.10 illustrate the scheduling results of the open-loop controller and the MPC controller. Comparing Fig. 5.9 and Fig. 5.10, the MPC controller reduces the possible delay resulting from the O_2^3 and therefore completes all jobs by rescheduling the operation of the machines.

Fig. 5.11 and Fig. 5.12 present the trajectory of the ASC of the open-loop controller and the MPC controller. Compared to Fig. 5.11, it can be seen from Fig. 5.12 that the MPC controller can adjust the speed and therefore reduce the possible delay of O_2^3 due to the unexpected slower speed of the ASC. This adjustment is based on the measurement of the location and the speed of ASC 2 which is processing O_2^3.

Table 5.7 compares the performances of the MPC controller and the open-loop controller. In the considered scenario, the MPC controller can reduce the possible delay and achieve lower energy consumption by rescheduling in a rolling horizon way in comparison to the open-loop controller.

Arrival of containers

Besides the uncertainties resulting from the operation delay, the precise time of arrival of new containers is regarded as another uncertainty that influences real-time control of the container handling. To test the performance of the proposed MPC controller, we set up a scenario in which a number of containers is being planned and the new container arrives as a disturbance. In this scenario, we assume 16 containers are operating and a new container arrives later as one example for illustrating how this disturbance is handled by the proposed MPC controller. It is assumed that the terminal is informed by the container arrives later at a particular time when the 16th containers leaves the vessel.

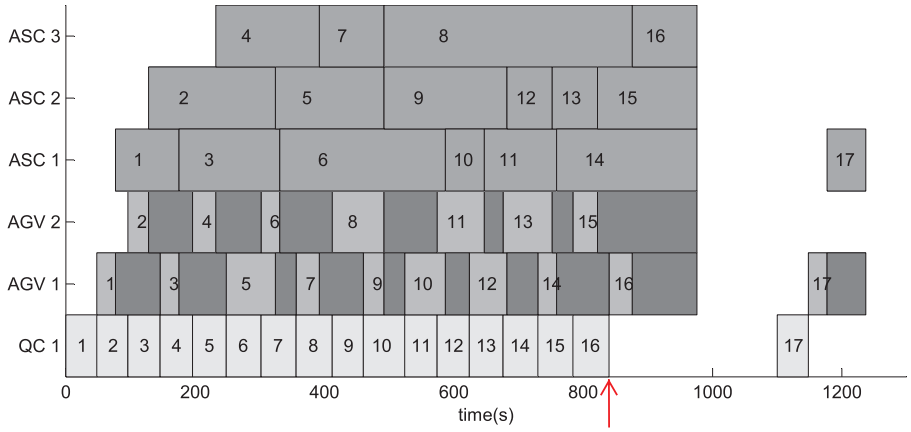


Figure 5.13: The scheduled result of the open-loop controller in the scenario of the new arrival containers.

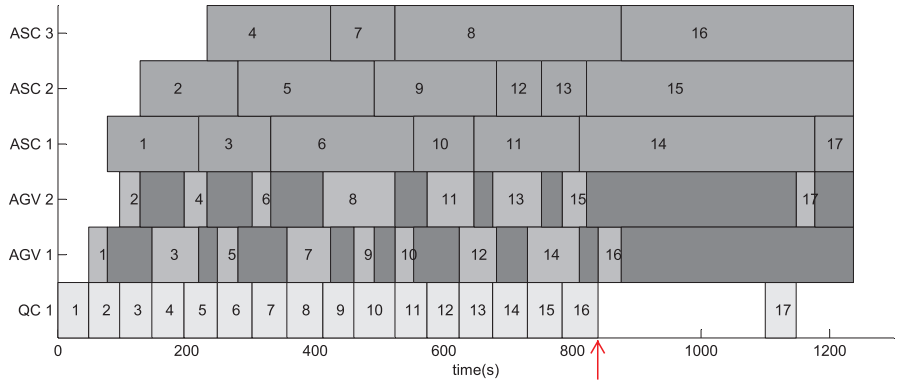


Figure 5.14: The scheduled result of the MPC controller in the scenario of the new arrival containers.

Fig. 5.13 and Fig. 5.14 present the scheduling result of the open-loop controller and the MPC controller both in the scenario of the new arrival container. The open-loop controller first plans the 16 containers completely and then plan the new arrival container when it just arrives at the terminal. The MPC controller schedules the 16 containers in a rolling horizon way and the rescheduling of the MPC controller is triggered when receiving the new container will arrive later. The open-loop controller does not consider the interaction of the ongoing jobs and the arrival job to be processed and therefore lose the opportunity of reducing energy consumption. Instead, when the new arrival container is informed, the MPC controller adjusts the operation times of ongoing jobs and the speed of the machines for the ongoing jobs for energy saving.

The performance of the MPC controller and the open-loop controller in the scenario of the new arrival container is computed in Table 5.8. It can be seen that the proposed MPC controller saves 16% of energy for the same completion time of all containers compared to

Table 5.8: The performance of controllers in the scenario of the new arrival container.

controller	MPC	Open-loop
Completion time	1237s	1237s
E_{tot}	8.82 kWh	10.40 kWh

the open-loop controller.

5.5 Concluding remarks

In this chapter, both discrete-event dynamics and continuous-time dynamics of a container handling system are considered. The discrete-event part is modeled as a three-stage hybrid flow shop and the continuous-time part is modeled as a double integrator. An event-driven receding horizon controller is proposed for improving energy-efficiency of the container handling system during the real-time operation. A proper prediction horizon N_{p} is chosen for achieving the competitive performance in a low computation burden. The energy-efficient container handling is implemented as a multi-objective optimization problem in the supervisory controller. The simulations indicate that the proposed controller obtains energy efficiency for the bay handling of the container vessel. The simulations also show the performance when the controller faces two types of uncertainties in real-time container handling.

Until now, the energy efficiency of the compact container terminal and the medium-size container terminal has been investigated, wherein the case of one QC is considered in these two types of terminal. In the case of one QC, only a small number of AGVs are involved and collision avoidance of AGVs are not considered. In the next chapter, the case of multiple QCs is considered, in which the research on collision-free trajectory planning of free-ranging AGVs integrated with scheduling of interacting machines will be carried out.

Chapter 6

Collision-free scheduling of free-ranging AGVs

Chapter 3-5 are concerned with a major problem of this thesis, i.e., to improve the energy efficiency of the operational control of automated container terminals, in the scope of the compact terminal and the medium-size terminal. The implementation of free-ranging AGVs has so far remained unclear for automated container terminals. On the one hand, collision-free trajectory planning of AGVs must be considered for safety reasons. On the other hand, the optimal coordination between AGVs and other types of equipment for high handling capacity is needed. The research problem of integrating collision-free trajectory planning of AGVs and the scheduling of interacting machines, which is considered for a large-size terminal, will be investigated in this chapter.

The research discussed in this chapter is based on [112, 113].

6.1 Introduction

This chapter investigates the integration of the collision-free trajectory planning of free-ranging AGVs with the scheduling of interacting machines in automated container terminals. As an emerging technology introduced in Chapter 1, the free-ranging AGV allows to move autonomously and therefore the path of the AGV is not fixed. This feature can shorten the driving distance considerably compared to the traditional mesh routing using a fixed path [31]. However, it is still not clear how to implement free-ranging AGVs in container terminals because the trajectory planning of AGVs is fairly complex. On the one hand collision avoidance of two AGVs must be considered for safety reasons, while AGVs cooperate with other types of machines (e.g., QCs and ASCs) interactively for loading or unloading vessels on the other hand. The operation times of AGVs can possibly be delayed due to collision avoidance, whereas the available literature [14, 17, 18, 59] cannot incorporate such disturbances for determining the scheduling of all pieces of equipment to be used.

To cope with this new problem, a sequential planning approach is proposed in this chapter. This sequential planning approach uses a hierarchical architecture for coordinating the

supervisory controller at a higher-level and a local controller of each piece of equipment at a lower level. The detailed schedule is determined job by job following a particular so-called overall graph sequence. For a particular job, the supervisory controller aims to optimally coordinate all pieces of equipment by determining the sequence of jobs for each particular piece of equipment and the time window during which each job is processed. The local controller determines the collision-free trajectory of AGVs taking into account the trajectories of other AGVs. Given the overall graph sequence, the actual time window of each job incorporating collision-free trajectories of AGVs is determined by solving a collection of mixed integer linear programming problems sequentially.

This chapter is organized as follows. Section 6.2 describes the discrete-event dynamics and the continuous-time dynamics of a typical automated container terminal. Section 6.3 proposes a sequential planning approach for generating the schedules, taking into account the collision-free trajectory planning. Section 6.4 illustrates the proposed trajectory planning approach in a number of representative simulations. Section 6.5 concludes this chapter in the end.

6.2 Modeling

As addressed in Chapter 4, container terminal dynamics can be considered as consisting of two levels. The higher-level consists of the discrete-event dynamics, while the lower-level consists of continuous-time dynamics. The actual operation of the machines is determined by the coordination of these two levels. Below the discrete-event model for the interaction of equipment and the continuous-time model for equipment will be discussed in detail.

6.2.1 Higher-level discrete-event dynamics

Chapter 4 presents the discrete-event dynamics of interacting machines for the case of one QC in the scope of the medium-size terminal. This chapter considers multiple QCs for a large-size terminal. For this, we extend the model in Chapter 4 and focus on the transition of a particular AGV from one QC to another possible QC. The extended definitions are illustrated in Fig. 6.1. Compared to Chapter 4, O_i^{22} considers P_j^2 as the destination for the transition to another possible QC that is to process job j.

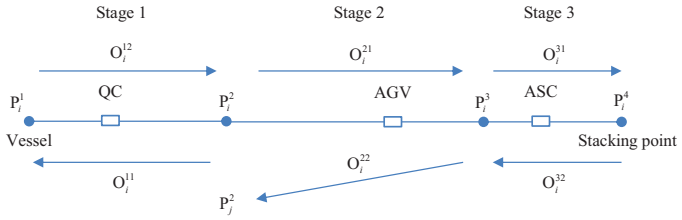


Figure 6.1: The sequence of transporting containers using three types of machines.

We next formalize the three stages and operations mathematically. Let there be N jobs of moving a container from vessel to stack. Similarly as in Chapter 4, the time constraints

on all jobs can then be described as follows:

$$a_i + R(1-\sigma_{0i}^1) \geq 0 \qquad \forall i \in \Phi \tag{6.1}$$
$$a_j + R(1-\sigma_{ij}^1) \geq b_i \qquad \forall i \in \Phi, \forall j \in \Phi \tag{6.2}$$
$$e_j + R(1-\sigma_{ij}^1) \geq b_i \qquad \forall i \in \Phi, \forall j \in \Phi \tag{6.3}$$
$$e_i + R(1-\sigma_{0i}^2) \geq 0 \qquad \forall i \in \Phi \tag{6.4}$$
$$b_i \geq a_i + t_i^{11} + t_i^{12} \qquad \forall i \in \Phi \tag{6.5}$$
$$b_i \geq e_i \qquad \forall i \in \Phi \tag{6.6}$$
$$b_i + t_i^{21} \leq c_i \qquad \forall i \in \Phi \tag{6.7}$$
$$c_i \leq d_i \qquad \forall i \in \Phi \tag{6.8}$$
$$c_j + R(1-\sigma_{ij}^3) \geq c_i + t_i^{31} + t_i^{32} \qquad \forall i \in \Phi, \forall j \in \Phi \tag{6.9}$$
$$b_j + t_j^{21} + R(1-\sigma_{ij}^3) \geq d_i + \Delta t \qquad \forall i \in \Phi, \forall j \in \Phi \tag{6.10}$$
$$e_j + R(1-\sigma_{ij}^2) \geq d_i - t_{ij}^{22} \qquad \forall i \in \Phi, \forall j \in \Phi \tag{6.11}$$
$$d_i - t_{ij}^{22} + R(1-\sigma_{ij}^2) \geq e_j \qquad \forall i \in \Phi, \forall j \in \Phi, \tag{6.12}$$

where, Φ, Φ_1 and Φ_2 are defined as in Chapter 4, and for $\forall\ i \in \Phi_1$ and $\forall\ j \in \Phi$ ($i \neq j$), we define decision variables

- σ_{ij}^1, σ_{ij}^2, σ_{ij}^3, a_i, b_i, c_i are the same defined in Chapter 4;
- d_i is the starting time of the transition from job i to job j in stage 2 , i.e., the time at which the AGV involved in job i and job j leaves P_i^3 for the next container from the QC;
- e_i is the arrival time of the AGV involved in job i in stage 1, i.e., the time at which this AGV is available at P_i^2;

and parameters

- t_i^{11}, t_i^{12}, t_i^{21}, t_i^{31} and t_i^{21} are the same as defined in Chapter 4;
- t_{ij}^{22} is the transition time of job i to job j in stage 2 processed by a particular AGV, which is defined particularly for the case of multiple QCs;
- R is a large positive number.

Inequalities (6.1) and (6.4) initialize the first job processed by a QC and an AGV, respectively. Inequality (6.2) describes the time relation among job i and j handled by the particular QC. Inequality (6.3) guarantees that at any time there is at most one AGV at the transfer point of each QC. Inequalities (6.5) and (6.6) indicate that job i starts in Stage 2 after it completes Stage 1. Inequalities (6.7) and (6.8) means job i starts in Stage 3 after it completes Stage 2. Inequality (6.9) represents the relation of job j and i handled by a particular ASC. Inequality (6.10) guarantees the time Δt is reserved between two successive jobs under the condition at any time there is at most one AGV at the transfer point of each stack. Inequalities (6.11) and (6.12) describe the transition of job i to job j processed by a particular AGV.

Table 6.1: The information related to operation $O_i^{h_1 h_2}$ in three stages.

Operation	Equipment	Starting time	Ending time	Processing time	Route
O_i^{11}	QC	a_i	$a_i + t_i^{11}$	t_i^{11}	$P_i^2 \rightarrow P_i^1$
O_i^{12}	QC	$a_i + t_i^{11}$	$a_i + t_i^{11} + t_i^{12}$	t_i^{12}	$P_i^1 \rightarrow P_i^2$
O_i^{21}	AGV	b_i	$b_i + t_i^{21}$	t_i^{21}	$P_i^2 \rightarrow P_i^3$
O_i^{22}	AGV	d_i	e_j	t_{ij}^{22}	$P_i^3 \rightarrow P_j^2$
O_i^{31}	ASC	c_i	$c_i + t_i^{31}$	t_i^{31}	$P_i^3 \rightarrow P_i^4$
O_i^{32}	ASC	$c_i + t_i^{31}$	$c_i + t_i^{31} + t_i^{32}$	t_i^{32}	$P_i^4 \rightarrow P_i^3$

Similarly as in Chapter 4, additional equality constraints of the discrete decision variables σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3 are described as (4.7) to (4.18).

Using the constraints (6.1)-(6.12) and (4.7)-(4.18), the discrete-event dynamics of the machines for the case of the multiple QCs are modeled as a three-stage hybrid flow shop. In the sequel, both the decision variables and parameters of this hybrid flow shop will be optimized interactively for the scheduling of the interacting machines and the detailed collision-free trajectory planning of AGVs.

Graph sequence

Determining the value of the decision variables of the discrete event dynamics considered in the earlier part of this section involves solving a hybrid flow shop scheduling problem in which the sequences of all jobs and time windows of all operations are determined simultaneously. In this, possible delays of AGVs are not taken into account. However, the operation of AGVs could be delayed due to collision avoidance. The determined time windows can then not directly be used for trajectory planning of AGVs. Therefore, a new approach for scheduling is proposed next that does take into account collision avoidance.

In this section we discuss the graph representation of the three-stage flow shop. We define a graph sequence in which the graph of all jobs to be processed can be seen. Furthermore, we propose a theorem that can be used to find a so-called overall graph sequence. This sequence enables generating the collision-free trajectories of AGVs sequentially and schedule all employed machines for transporting all containers.

The three-stage flow shop described in the earlier part of this section can be represented by three graphs in which each graph represents the jobs processed in a particular stage in a particular sequence. The integer variables σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3 define the sequence in which jobs are processed by a particular piece of equipment in Stage 1, Stage 2 and Stage 3. The values of σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3 can be represented as the graph of each respective stage. Fig. 6.2 gives an example of processing 6 jobs (2 QCs, 3 AGVs and 3 ASCs) to illustrate the possible job sequence per piece of equipment per stage. In Stage 1, job 1, 2 and 3 are processed sequentially by one QC and job 4, 5 and 6 are processed sequentially by another QC. Job 1 and job 4 are the first jobs processed by the two QCs, and job 3 and job 6 are the last jobs processed by the two QCs. The graph of Stage 2 and Stage 3 can be explained similarly.

We define a graph sequence Q as the graph representing all jobs processed in a particular stage of the hybrid flow shop considered in Section 2.1. More precisely, for Stage 1, Q is defined as $Q = \{q_n\}_1^N = \{...,i,...,j,...\}(i \in \Phi, j \in \Phi, i \neq j)$, where job j is processed after

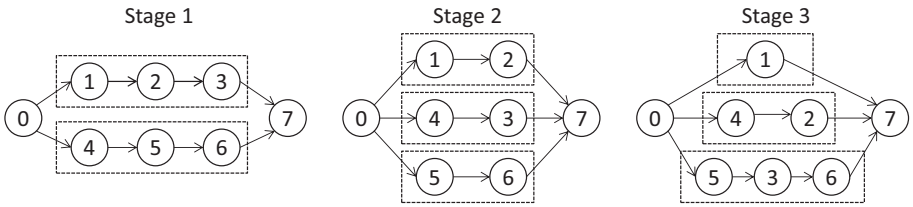


Figure 6.2: Illustration of possible job sequence decisions in three stages for processing 6 jobs (0 and 7 represent dummy jobs; different paths correspond to different pieces of equipment).

job i. For job i and job j, if job j is processed directly after job i and job i and job j are processed by a particular piece of equipment of the graph then $\sigma_{ij}^1 = 1$, otherwise $\sigma_{ij}^1 = 0$. Similar descriptions can be made for Stage 2 and Stage 3. A possible graph sequence for Stage 1 is $\{1,4,2,5,6,3\}$. For instance, job 2 is processed directly after job 1 by the same piece of equipment. However, job 2 is not processed after job 4 since job 2 and job 4 are not processed by the same piece of equipment. For a particular stage, there can be multiple graph sequences. Another graph sequence for Stage 1 is $\{4,1,2,5,6,3\}$.

With a graph sequence for a particular stage, all jobs of the stage can be planned individually and the disturbance of the operation for each job in each stage can be incorporated when a particular job is planned. This allows to plan all jobs of the stage following this particular sequence. Particularly for Stage 2, the longer processing time resulting from the disturbance of AGVs for collision avoidance can be incorporated in the planning of all jobs following the graph sequence.

For sequentially planning all jobs of this hybrid flow shop an overall graph sequence for each of three stages is required. However, just combining feasible graph sequences for individual stages may not lead to an overall feasible graph sequence. A graph sequence both for Stage 1 and Stage 2 can be found easily, but this graph sequence can be conflicting with the graph of Stage 3. For instance, $\{1,4,2,5,6,3\}$ is a feasible graph sequence for Stage 1 and also for Stage 2; however it is not a graph sequence for Stage 3. This is because in the graph sequence $\{1,4,2,5,6,3\}$ job 3 is processed after job 6, while job 6 is processed after job 3 in Stage 3 of Fig. 6.2 resulting in a conflict. Job 6 is planned before job 3 using the graph sequence $\{1,4,2,5,6,3\}$, whereas in Stage 3 the starting time of job 3 is planned before the starting time of job 6. Due to a possible longer processing time of job 3 in Stage 2 for collision avoidance, a particular ASC does not have enough time for processing job 3 and inequality (9) cannot hold. Therefore, an overall graph sequence for all stages is needed for planning of all jobs when incorporating the disturbance of AGVs for collision avoidance. For the above example, a feasible overall graph sequence is $\{1,4,5,2,3,6\}$.

To obtain an overall graph sequence, we consider the continuous decision variables $b_i + t_i^{21}$ ($i \in \Phi$), sorted based on their values. The indices of this sorted sequence give the order of the jobs. Each job of the overall graph sequence is related to the value of these variables. Naturally a sequence can be determined sorting the values in an ascending way. In the following we consider to obtain the overall graph sequence sorting the values of $b_i + t_i^{21}$.

We propose a theorem in which a conditional constraint is stated that needs to be satis-

fied for obtaining a feasible overall graph sequence:

Theorem 6.1 *For the three-stage flow shop (*(6.1) - (6.12) *and (4.7)-(4.18)), if* $R(1-\sigma_{ij}^1)+b_j+t_j^{21} > b_i+t_i^{21}$*, then the sorted sequence* $Q=\{...,i,...,j,...\}$ $(b_j+t_j^{21} \geq b_i+t_i^{21}, i \in \Phi, j \in \Phi, i \neq j)$ *is a feasible overall graph sequence.*

Proof : Considering the (6.7), (6.8) and (6.10), we have that $R(1-\sigma_{ij}^3)+b_j+t_j^{21} > d_i \geq b_i+t_i^{21}$, so $R(1-\sigma_{ij}^3)+b_j+t_j^{21} > b_i+t_i^{21}$.
Since for a particular AGV job j is directly processed after job i, we have $R(1-\sigma_{ij}^2)+b_j+t_j^{21} > b_i+t_i^{21}$.

Therefore, for the sequence $Q=\{...,i,...,j,...\}$ $(b_j+t_j^{21} \geq b_i+t_i^{21}, i \in \Phi, j \in \Phi, i \neq j)$, $R(1-\sigma_{ij}^1)+b_j+t_j^{21} > b_i+t_i^{21}$, $R(1-\sigma_{ij}^2)+b_j+t_j^{21} > b_i+t_i^{21}$ and $R(1-\sigma_{ij}^3)+b_j+t_j^{21} > b_i+t_i^{21}$. This means that Q is a feasible overall graph sequence. □

This overall graph sequence Q guarantees for a particular piece of equipment in each stage that the start of job j is always after the completion of job i if $\sigma_{ij}^1=1$, $\sigma_{ij}^2=1$ or $\sigma_{ij}^3=1$, even if the completion of job i is delayed due to dynamical collision avoidance. This property will be used below for determining the collision-free trajectory of AGVs and generating the schedule of the three-stage flow shop.

6.2.2 Dynamical model of equipment

This section presents the continuous-time dynamics of the pieces of equipment (QCs, AGVs and ASCs). In Chapter 4, the trajectories of the QC, the AGV and the ASC are all modeled in one dimension for a medium-size terminal, in which the collision avoidance of AGVs is not considered. This chapter considers a large-size terminal where a two-dimensional vehicle dynamics and two types of obstacles are modeled. In this chapter, for the QC and the ASC we still consider one dimensional model similarly as presented in Chapter 4.

Dynamical model of AGVs

In container terminals, AGVs are employed to transport containers between the quayside area and the stacking area. We consider the common situation of a homogeneous fleet of AGVs, i.e., the dynamics of the AGVs are identical. For each AGV, a point-mass model is used to approximate the dynamical behavior in two-dimensional space as follows: at time instant k

$$\begin{bmatrix}\mathbf{r}_p(k+1)\\ \mathbf{v}_p(k+1)\end{bmatrix}=\begin{bmatrix}I_2 & \Delta t I_2\\ 0_2 & I_2\end{bmatrix}\begin{bmatrix}\mathbf{r}_p(k)\\ \mathbf{v}_p(k)\end{bmatrix}+\begin{bmatrix}0.5(\Delta t)^2 I_2\\ \Delta t I_2\end{bmatrix}\mathbf{u}_p(k), \tag{6.13}$$

where AGV p has a position $\mathbf{r}_p(k)=\left[r_p^{\mathrm{x}}(k)\ r_p^{\mathrm{y}}(k)\right]^{\mathrm{T}}$ $(\mathbf{r}_p(k) \in \mathbb{R}^2)$ and a velocity $\mathbf{v}_p(k)=\left[v_p^{\mathrm{x}}(k)\ v_p^{\mathrm{y}}(k)\right]^{\mathrm{T}}$ $(\mathbf{v}_p(k) \in \mathbb{R}^2)$. Each AGV is assumed to respond to control actions $\mathbf{u}_p(k)=\left[u_p^{\mathrm{x}}(k)\ u_p^{\mathrm{y}}(k)\right]^{\mathrm{T}}$ $(\mathbf{u}_p(k) \in \mathbb{R}^2)$.

Constraints on velocity and action are described as follows: $\forall p \in [1,...,n_{\mathrm{agv}}]$

$$v_p^{\mathrm{x}}(k)^2+v_p^{\mathrm{y}}(k)^2 \leq {v_{\max}}^2 \tag{6.14}$$

$$u_p^x(k)^2 + u_p^y(k)^2 \leq {u_{\max}}^2, \tag{6.15}$$

where $u_{\max}$ and $v_{\max}$ are limits on the acceleration and velocity. These nonlinear constraints could result in time-consuming computation. To avoid this potential, this velocity constraint is approximated by polygons using linear equalities [83], as follows: $\forall m \in [1,...,M]$

$$v_p^x(k)\sin(\frac{2\pi m}{M}) + v_p^y(k)\cos(\frac{2\pi m}{M}) \leq v_{\max} \tag{6.16}$$

$$u_p^x(k)\sin(\frac{2\pi m}{M}) + u_p^y(k)\cos(\frac{2\pi m}{M}) \leq u_{\max}, \tag{6.17}$$

where M is an arbitrary number used to achieve better approximations (large M results in a better approximation).

The approximation of maximal velocities by polygons, compared with the exact constraint (the circle), is illustrated in Fig. 6.3. We choose to approximate the constraints of the speed and the control variables with $M = 10$ (as suggested by [83]).

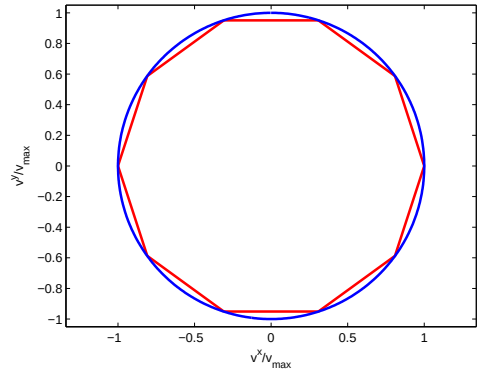


Figure 6.3: The approximation of the velocity limit of AGVs.

Collision avoidance

Two types of collisions can occur regarding AGVs. The first one is a collision with a static obstacle of the stacking area near the transfer point. There are two tracks of a stacking crane on one side of the stack where containers are handled. For security reasons, AGVs should not approach the area of these tracks. The other possibility is a collision with another moving AGV. To avoid both types of the collision safely, we consider that each AGV has a rectangle safety zone, as suggested by [83]. This rectangle zone has the safety distance d with the area $(2d \times 2d)$ as shown in Fig. 6.4.

- Static obstacle

  Fig. 6.5 illustrates the static obstacle considered. The static obstacle can be formally described as rectangular area illustrated in Fig. 6.5(b) (see [88]). The rectangular area can be described by the lower left corner $(s^{\text{low,x}}, s^{\text{low,y}})$ and upper right corner $(s^{\text{high,x}}, s^{\text{high,y}})$. To avoid the static obstacle, the AGV must stay out of this rectangular

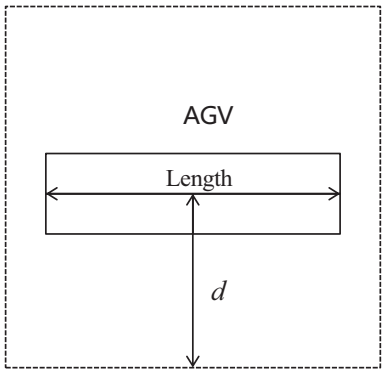
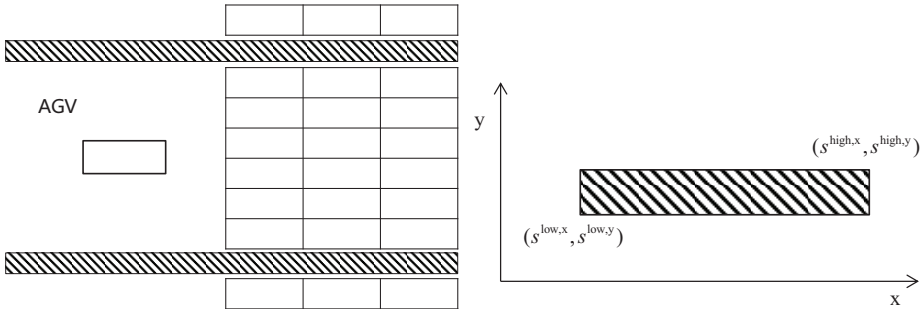


Figure 6.4: The safety zone of an AGV.



(a) Two static obstacle areas near the stacking area. (b) The coordinates of the static obstacle area.

Figure 6.5: Illustration of the static obstacle area.

area. This avoidance requirement can be described using the following constraints: $\forall p \in [1,...,n_{\text{agv}}]$,

$$\begin{aligned} r_p^{\text{x}}(k) &\leq s^{\text{low,x}} - d \\ \text{or } r_p^{\text{x}}(k) &\geq s^{\text{high,x}} + d \\ \text{or } r_p^{\text{y}}(k) &\leq s^{\text{low,y}} - d \\ \text{or } r_p^{\text{y}}(k) &\geq s^{\text{high,y}} + d \end{aligned} \tag{6.18}$$

By introducing binary variables, the or-constraint (6.18) can be written in the following and-constraint form, which is standard for optimization problem formulations:

$$\begin{aligned} r_p^{\text{x}}(k) &\leq s^{\text{low,x}} - d + Rb_{\text{in},1} \\ r_p^{\text{x}}(k) &\geq s^{\text{high,x}} + d - Rb_{\text{in},2} \\ r_p^{\text{y}}(k) &\leq s^{\text{low,y}} - d + Rb_{\text{in},3} \\ r_p^{\text{y}}(k) &\geq s^{\text{high,y}} + d - Rb_{\text{in},4} \end{aligned} \tag{6.19}$$

$$\sum_{\tau=1}^{4} b_{\text{in},\tau} \leq 3, \forall p \in [1,...,n_{\text{agv}}], \tag{6.20}$$

where R is a large positive number and $b_{\text{in},\tau}$ is the binary variable for modeling the static obstacle, constraints (6.17) and (6.18) ensure that at least one of the equalities in (6.16) is satisfied, which ensures that the AGV is out of the static obstacle zone.

- Moving obstacle

  In the case when multiple AGVs are transporting containers to different destinations, collisions between vehicles also need to be considered. At each time step every pair of vehicles p_1 and p_2 must be a minimal distance apart from each other in a 2 dimensional environment. Due to the safety distance d for each AGV, $2d$ is the distance that 2 AGVs need to be apart at least. The collision avoidance constraints can thus be described as follows:

$$\|r^{\text{x}}_{p_1}(k) - r^{\text{x}}_{p_2}(k)\| \geq 2d \text{ or } \|r^{\text{y}}_{p_1}(k) - r^{\text{y}}_{p_2}(k)\| \geq 2d, \tag{6.21}$$

  $\forall p_1 \in [1,...,n_{\text{agv}}, p_2 \in [1,...,n_{\text{agv}}]$ with $p_1 \neq p_2$,

  where $\mathbf{r}_{p_1}(k)$ ($\mathbf{r}_{p_1}(k) = [r^{\text{x}}_{p_1}(k), r^{\text{y}}_{p_1}(k)]$) and ($\mathbf{r}_{p_2}(k) = [r^{\text{x}}_{p_2}(k), r^{\text{y}}_{p_2}(k)]$) are the positions of vehicle p_1 and p_2 at step k, respectively.

  Similarly as for transforming (30) into standard optimization format, constraint (6.21) is written using binary variables as follows:

$$\begin{aligned} r^{\text{x}}_{p_1}(k) &\leq r^{\text{x}}_{p_2}(k) - 2d + Rb_{\text{m},1}(k) \\ r^{\text{x}}_{p_1}(k) &\geq r^{\text{x}}_{p_2}(k) + 2d - Rb_{\text{m},2}(k) \\ r^{\text{y}}_{p_1}(k) &\leq r^{\text{y}}_{p_2}(k) - 2d + Rb_{\text{m},3}(k) \\ r^{\text{y}}_{p_1}(k) &\geq r^{\text{y}}_{p_2}(k) + 2d - Rb_{\text{m},4}(k) \end{aligned} \tag{6.22}$$

$$\sum_{\tau=1}^{4} b_{\text{m},\tau}(k) \leq 3, \forall k, \tag{6.23}$$

  where $b_{\text{m},\tau}$ is the binary variable for modeling the moving obstacle and R is a large positive number for modeling the relaxation of inequality in (6.22) when $b_{\text{m},\tau} = 1$ and vice versa. Equations (6.22) and (6.23) ensure that (6.21) is satisfied.

6.3 Hierarchical controller architecture

This chapter focuses on the collision-free trajectory determined in Stage 2 integrated with scheduling of three stages. This is highlighted in the hierarchical control architecture of the container terminal as shown in Fig. 6.6. Details of the stage controller and the lower-level controller for the QCs and the ASCs can be found in Chapter 4.

Two levels of the hierarchical control architecture in particular for Stage 2 are discussed in the following:

- The higher level

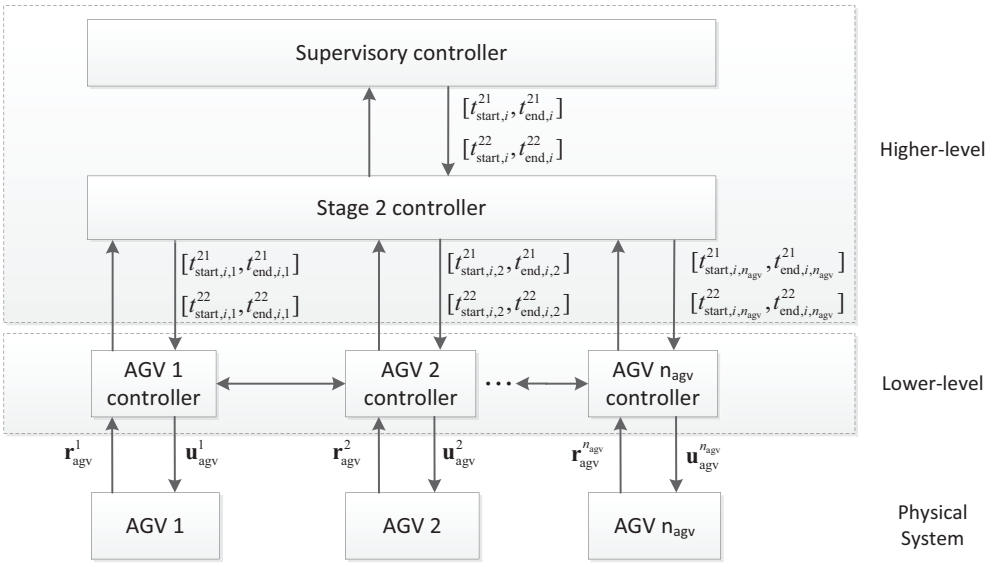


Figure 6.6: The hierarchical control architecture.

The higher level controller consists of a supervisory controller and a stage controller for each stage. The supervisory controller coordinates three stages of machines and schedules time windows of processing operations in each stage by means of determining the sequence of jobs. The stage controller assigns the time window of each operation to a particular machine.

- The lower level

  At the lower level, the system is driven by the continuous-time dynamics of each machine. In particular in Stage 2, the continuous trajectory generation of an AGV involves collision avoidance with request to other AGVs and static obstacles. After receiving the starting point for performing a particular operation from the higher level, the lower-level controller of an AGV aims to complete its operation, taking into account the static obstacle and dynamical obstacles. Based on the continuous-time dynamics and constraints for collision avoidance, and using a certain cost function, an optimal control problem is formulated and solved. The obtained trajectory of the AGV is used for collision-free trajectory planning of other AGVs. The actual completion time of the AGV is sent back to higher-level controller.

The overall structure for the interaction of the two-level controllers are described in Fig. 6.7.

6.3.1 Supervisory controller

The supervisory controller aims to determine the sequence of jobs in each stage that minimizes the makespan. Here we define the makespan as the completion time of the last container that leaves the vessel when both the QC and the ASC are available. In other

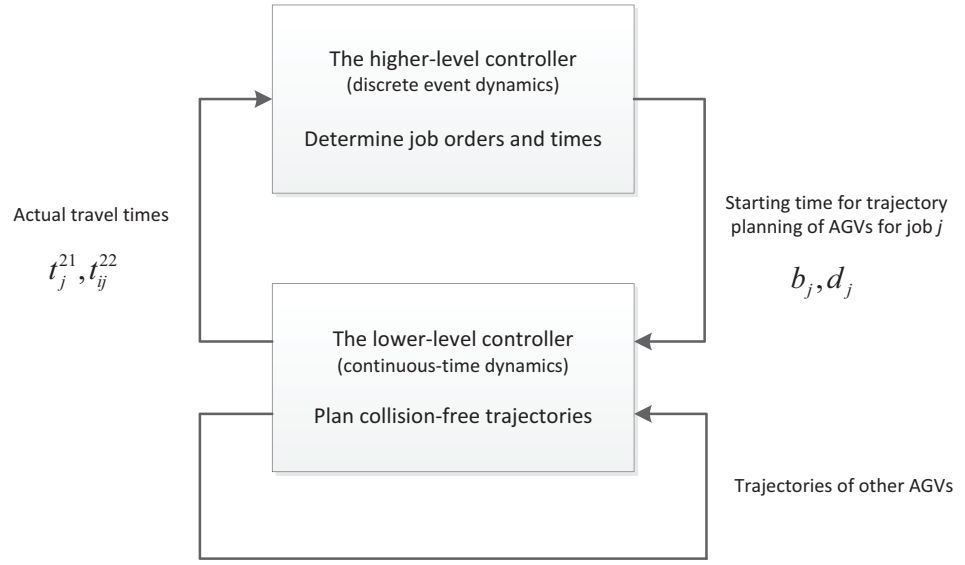


Figure 6.7: The overall structure of the two level controller.

words, it is defined as max $\{a_1+t_1^{11}+t_1^{12},...,a_N+t_N^{11}+t_N^{12},e_1,...,e_N\}$, i.e., $\|\mathbf{w}\|_\infty$, where $\mathbf{w}=\left[a_1+t_1^{11}+t_1^{12},a_2+t_2^{11}+t_2^{12},\ldots,a_N+t_N^{11}+t_N^{12},e_1,e_2,\ldots,e_N\right]^{\mathrm{T}}$ and $\|\cdot\|_\infty$ denotes the infinity norm.

In this chapter, we use the overall graph sequence to determine the overall planing problem job by job sequentially. When the operations with respect to a particular job are planned, for the all remaining jobs, the supervisory controller determines the integer decision variables related to the job sequences (σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3) and the continuous variables related to the time windows (a_i, b_i, c_i, d_i and e_i) simultaneously, considering the parameters ($t_i^{h_1h_2}$ and t_{ij}^{22}) in the case of no avoidance. These variables are obtained solving a hybrid flow shop scheduling problem, resulting in the job to be planned following the overall graph sequence. Then for this particular job, the parameters $t_i^{h_1h_2}$ and t_{ij}^{22} are obtained as the result of the detailed collision-free trajectory planning of a particular AGV.

For determining the orders of jobs in all stages of the hybrid flow shop problem, we add the additional constraint discussed in Theorem 1. For the sake of notation compactness, we define:
$\mathbf{a}=\left[a_1,a_2,\cdots,a_N\right]^{\mathrm{T}}$, $\mathbf{b}=\left[b_1,b_2,\cdots,b_N\right]^{\mathrm{T}}$, $\mathbf{c}=\left[c_1,c_2,\cdots,c_N\right]^{\mathrm{T}}$, $\mathbf{d}=\left[d_1,d_2,\cdots,d_N\right]^{\mathrm{T}}$, $\mathbf{e}=\left[e_1,e_2,\cdots,e_N\right]^{\mathrm{T}}$, σ is defined the same as in Chapter 4.

The scheduling problem can be then rewritten as follows:

$$\min_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{e},\sigma} \|\mathbf{w}\|_\infty \tag{6.24}$$

subject to

$$R(1-\sigma_{ij}^1)+b_j+t_j^{21}>b_i+t_i^{21},\quad \forall i\in\Phi,\forall j\in\Phi \tag{6.25}$$

and (6.1) - (6.12) and (4.7) - (4.18). This optimization problem is a mixed-integer linear programming problem.

Solving the hybrid flow shop problem is computationally expensive due to its NP-hardness [86]. For speeding up the computation, we consider adding the sum of operations in Stage 1 as a lower bound of the objective function.

Note that the job sequences (σ_{ij}^1, σ_{ij}^2 and σ_{ij}^3) do not change when a particular job to be planned is considered for sequentially planning all jobs. However, for the job i, O_i^{21} and O_i^{22}, which are associated with the collision-free trajectory planning, are interdependent. Between the transition O_i^{21} and O_i^{22}, the starting time of O_i^{21} (b_i) or O_i^{22} (d_i) depends on the coordination of pieces of equipment that process job i. This coordination is carried out by the higher-level controller using the following optimization problem:

$$\min_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{e}} \|\mathbf{w}\|_\infty \tag{6.26}$$

subject to (6.1) - (6.12) and (4.7) - (4.18).

This optimization problem is a linear programming problem for which the solution can be obtained fast by existing solvers.

6.3.2 Stage controller

Given a sequence of jobs to be determined, the aim of the stage controller is to assign each job to a specific machine in each stage. In the hybrid flow shop problem this assignment is implicitly expressed in (4.7) - (4.18) without pointing out the exact machine. In the following part, we define three mapping functions for describing the explicit assignment.

Considering all AGVs are identical in Stage 2, there is no difference between the choice of AGVs. We define $\Psi_{\text{agv}} = \{1, 2, ..., n_{\text{agv}}\}$ representing the set of AGVs.

$$f_{\text{agv}} : \Phi \to \Psi_{\text{agv}}, \tag{6.27}$$

where f_{agv} is a function that maps the set of jobs Φ to the set of AGVs Ψ_{agv}. $f_{\text{agv}}(i)$ describes the particular AGV assigned to job i. The time windows of job i assigned to a particular AGV can be denoted by $[t^{21}_{\text{start},i,f_{\text{agv}}(i)}, t^{21}_{\text{end},i,f_{\text{agv}}(i)}]$ and $[t^{22}_{\text{start},i,f_{\text{agv}}(i)}, t^{22}_{\text{end},i,f_{\text{agv}}(i)}]$.

Regarding the QC and the ASC, the mapping of job i to the machine is predetermined since each container has a certain origin in the vessel and a certain destination in the stack. Here we use $f_{\text{qc}}(i)$ and $f_{\text{asc}}(i)$ for describing the assigned QC and the assigned ASC for job i. Then the time windows of job i assigned to a particular QC is denoted by $[t^{11}_{\text{start},i,f_{\text{qc}}(i)}, t^{11}_{\text{end},i,f_{\text{qc}}(i)}]$ and $[t^{12}_{\text{start},i,f_{\text{qc}}(i)}, t^{12}_{\text{end},i,f_{\text{qc}}(i)}]$. Similarly, the time windows of job i assigned to a particular ASC are denoted by $[t^{31}_{\text{start},i,f_{\text{asc}}(i)}, t^{31}_{\text{end},i,f_{\text{asc}}(i)}]$ and $[t^{32}_{\text{start},i,f_{\text{asc}}(i)}, t^{32}_{\text{end},i,f_{\text{asc}}(i)}]$.

6.3.3 Lower-level controller

In this section, we consider a minimal-time control problem for the lower-level controller of the AGV. By solving this control problem, the lower-level controller of the AGV generates a collision-free trajectory. In this minimal-time problem, the AGV is required to complete the operation O_i^{21} or O_i^{22} as fast as possible, considering the static obstacles and the dynamical obstacles. We use a numerical approach for calculating the minimal time required by a particular AGV to complete the operation moving from origin $\mathbf{r}_0$ to destination $\mathbf{r}_f$ subject to its dynamics and constraints on velocity, acceleration and obstacle. Here, $\mathbf{r}_0$ and $\mathbf{r}_f$ are chosen from one of the transfer points for the QC and the ASC for job i.

Table 6.2: The choice of origin and destination with respect to job i.

Origin	Destination	Operation	Operation time	Starting time
QC $f_{\text{qc}}(i)$	Stack $f_{\text{asc}}(i)$	O_i^{21}	t_i^{21}	$t^{21}_{\text{start},i,f_{\text{agv}}(i)}$
Stack $f_{\text{asc}}(i)$	QC $f_{\text{qc}}(j)$ with $y_{ij}=1$	O_i^{22}	t_{ij}^{22}	$t^{22}_{\text{start},i,f_{\text{agv}}(i)}$

The exact choices of origin and destination with respect to job i are given in Table 6.2. This choice depends on the assignment of the QC $f_{\text{qc}}(i)$ and the ASC $f_{\text{asc}}(i)$ when job i is transported from the quayside to the stack. Also, the origin and the destination rely on $f_{\text{asc}}(i)$ and $f_{\text{qc}}(j)$ when AGV moves from the stack to the quayside for picking up job j after job i considering $\sigma_{ij}^2 = 1$.

Suppose T is the length of the given time window. Within a given interval $[0,...,T-1]$, AGV p reaches $\mathbf{r}_f$ only at a certain moment, which is modeled by a binary variable at time k ([83]). This constraint can be represented as follows:

$\forall k \in [1,...,T-1], \forall p \in [1,...,n_{\text{agv}}]$,

$$\begin{aligned} r^{\text{x}}(k) - r_f^{\text{x}} &\le R(1-b_p(k)) \\ r^{\text{x}}(k) - r_f^{\text{x}} &\ge -R(1-b_p(k)) \\ r^{\text{y}}(k) - r_f^{\text{y}} &\le R(1-b_p(k)) \\ r^{\text{y}}(k) - r_f^{\text{y}} &\ge -R(1-b_p(k)) \end{aligned} \tag{6.28}$$

$$\sum_{k=1}^{T-1} b_p(k) = 1, \forall k \in [1,...,T-1], \tag{6.29}$$

where $b_p(k) \in \{0,1\}$ is a binary variable, R is a large positive number to guarantee the constraints in (6.28) are active only when $b_p(k)=1$. Equations (6.23) and (6.29) force the position $\mathbf{r}(k)$ of AGV p to reach the target $\mathbf{r}_f$ with condition $b_p(k)=1$.

If we define $t(k)$ as the elapsed time at time k ($t(k)=k$), then $t(k)b_p(k)$ is the finishing time when $b_p(k)=1$. The minimal time from the origin and the destination can be obtained by minimizing the sum of finishing times. Here we also consider improving energy efficiency of the planned trajectory and minimizing the sum of accelerations with a small penalty λ_{eng} in the objective function. Therefore, this minimal-time optimization problem can be formulated as follows:

$$\min_{\mathbf{u},\mathbf{b}} \sum_{k=1}^{T-1} t(k)b_p(k) + \lambda_{\text{eng}} \sum_{k=0}^{T-1} \left(\left| u_p^x(k) + u_p^y(k) \right| \right), \tag{6.30}$$

subject to (6.13)-(6.17), (6.22)-(6.23) and (6.28)-(6.29),
where $\mathbf{u} = [u_p^x(0), u_p^y(0), u_p^x(1), u_p^y(1), ..., u_p^x(T-1), u_p^y(T-1)]^{\text{T}}$ denotes continuous decision variables and $\mathbf{b} = [b_p(0), b_p(1), ..., b_p(T-1)]^{\text{T}}$ denotes binary decision variables.

The optimization problem formulated above is an MILP problem. The result of this optimization problem will be used for updating t_i^{21} and t_{ij}^{22}, and therefore determine the starting time of the next operation for an AGV.

Algorithm 2 Generate collision-free trajectory

The supervisory controller determines σ, **a**, **b**, **c**, **d** and **e**
The supervisory controller determines Q
The stage controller assigns f_{qc} f_{agv} and f_{asc}
for $n = 1 : N$ **do**
  $j = q_n$
  if $\sigma^2_{0j} = 1$ **then**
    The local controller receives b_j and computes t^{21}_j
    The local controller computes t^{21}_j
    The supervisory controller updates t^{21}_j, b_j and c_j
  else
    $\forall i, j$ $\sigma^2_{ij} = 1$ the local controller receives d_i and computes t^{22}_{ij}
    The supervisory controller updates t^{22}_{ij} and b_j
    The local controller receives b_j and computes t^{21}_j
    The supervisory controller updates t^{21}_j, b_j and c_j
  end if
  The supervisory controller solves the hybrid flow shop scheduling problem for the remaining jobs
  The supervisory controller updates Q
end for

6.3.4 Algorithm for collision-free scheduling

Algorithm 2 presents the key procedures for solving the overall control problem, in which the collision-free scheduling is planned sequentially following the overall graph sequence job by job. For the planning of all operations related to a particular job, two levels of controllers coordinate with each other. For the initial solution, the supervisory controller determines the order of jobs and the related operation times considering the expected completion time for each operation, which is obtained solving a three-stage hybrid flow shop problem. Therefore, the overall graph sequence can be obtained and the first job to be planned is given. Then the time windows for processing operations in each stage are sent from the supervisory controller to the stage controller. The stage controller assigns each operation to a particular machine used in the that stage. For the operations associated with the first job to be planned, these steps provide preliminary inputs for the operations associated with the first job to be planned, including the collision-free trajectory planning of AGVs.

If job j is the first job processed by a particular AGV, only operation O^{21}_j is considered. For the trajectory planning of O^{21}_j, the assigned AGV $f_{\text{agv}}(j)$ receives the starting time b_j (i.e., $t^{21}_{\text{start},j,f_{\text{agv}}(j)}$) from the higher level controller. When the collision-free trajectory for O^{21}_j is complete, the lower-level controller of AGV O^{21}_j sends O^{21}_j back to the higher-level controller. As a result, c_j for the operations of the ASC and the variables for the other unplanned jobs can be updated in the hybrid flow shop.

Alternatively, if job j is not the first job processed by a particular AGV, we consider both the operation O^{22}_{ij} and O^{21}_j due to the transition from job i to job j ($\sigma^2_{ij} = 1$). Operation O^{22}_{ij} starts from $t^{22}_{\text{start},i,f_{\text{agv}}(i)}$ given by the higher-level controller. When the collision-free

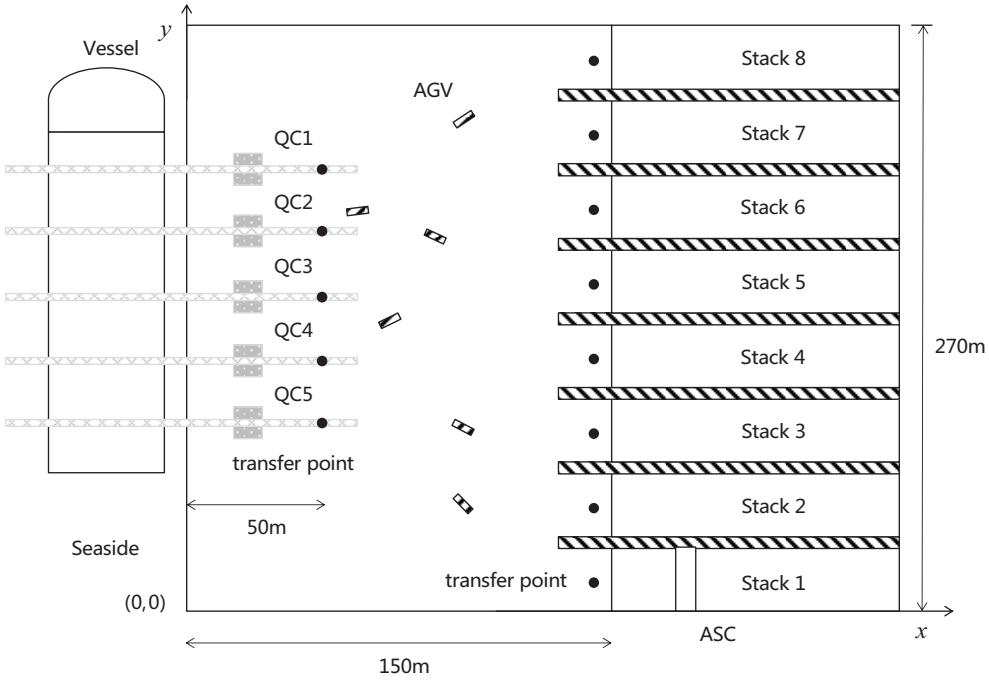


Figure 6.8: Benchmark System 3-the large-size terminal.

trajectory for operation O_{ij}^{22} is complete, t_{ij}^{22} (i.e., $t_{\text{end},i,f_{\text{agv}}(i)}^{22}$) is sent back to the higher-level controller for updating b_j that is used as the starting time for O_j^{21}. Then the local controller receives b_j as the starting time of O_j^{21} ($t_{\text{start},j,f_{\text{agv}}(j)}^{21}$) for generating the collision-free trajectory for operation O_j^{21}. The state updates for operation O_j^{21} are the same as mentioned in the case when job j is the first job processed by a particular AGV.

When the related operations with respect to job j are planned, the related variables will be fixed for determining the variables of the remained jobs to be planned. As a result, the overall graph sequence S will be updated as well.

In summary, collision-free trajectory planning for all jobs in Stage 2 can be carried out sequentially following overall graph sequence S. Since there is no collision avoidance considered for Stage 1 and Stage 3, the time windows of operations in Stage 1 and Stage 3 are updated automatically when the trajectory-free trajectory planning is generated sequentially.

6.4 Simulation experiments

In this section, the performance of the proposed approach for generating collision-free scheduling of free-ranging AGVs will be evaluated and discussed. For the large-size container terminal, Benchmark System 3 proposed in Chapter 2 is considered in this chapter, which is shown in Fig. 6.8. Since this chapter focuses on the high handling capacity of the container handling system and the collision-free trajectory planning of free-ranging AGVs. The completion time, the computation time, the driving distance and the relative distance,

Table 6.3: The coordinates of transfer points in the benchmark system.

Location	Coordinate (x,y)	Location	Coordinate (x,y)
QC1	(50,170)	Stack 1	(145,12.5)
QC2	(50,150)	Stack 2	(145,47.5)
QC3	(50,130)	Stack 3	(145,82.5)
QC4	(50,110)	Stack 4	(145,117.5)
QC5	(50,90)	Stack 5	(145,152.5)
		Stack 6	(145,187.5)
		Stack 7	(145,222.5)
		Stack 8	(145,257.5)

Table 6.4: The configuration of all simulation experiments.

Case	# containers	# AGVs	# ASCs	# QCs
3QC-1 $\sim$ 3QC-5	24	6	5	3
4QC-1 $\sim$ 4QC-5	32	8	6	4
5QC-1 $\sim$ 5QC-5	40	10	8	5

which are proposed in Chapter 2 will be used as KPIs in this chapter.

6.4.1 Setup

In this chapter, we consider scenarios with 3 QCs, 4 QCs and 5 QCs for unloading a vessel. The number of AGVs is twice the number of QCs for supporting a high utilization of the QC as considered in Chapter 4. The number of ASCs is considered to be consistent with the options in [18, 62]. The detailed configuration of the number of employed pieces of equipment are shown in Table. 6.4. The settings of this benchmark system are presented as follows:

- The initial position of all QCs is set to their unloading position. The initial position of all AGVs is set to their loading position. The initial position of all ASCs is set to their loading position;
- Eight containers are considered for each QC with the same arrival time, which is the same as considered in Chapter 4;
- The processing time of one container by the QC depends on the specific position in the vessel;
- Each container here considered has the same vertical position in the vessel;
- The storage location of each container to be transported is generated randomly;
- One ASC is installed in one stack and the containers stored in each stack have different storage places;

Table 6.5: The initial result of the hybrid flow shop scheduling.

Case	makespan	computation time	optimality
3QC-1	412s	787s	Yes
3QC-2	412s	6656s	Yes
3QC-3	412s	2028s	Yes
3QC-4	412s	25220s	Yes
3QC-5	412s	33014s	Yes
4QC-1	423s	10h	No
4QC-2	429s	10h	No
4QC-3	418s	10h	No
4QC-4	427s	10h	No
4QC-5	452s	10h	No
5QC-1	450s	10h	No
5QC-2	475s	10h	No
5QC-3	430s	10h	No
5QC-4	428s	10h	No
5QC-5	420s	10h	No

- The service time of the QC, the AGV and the ASC for exchanging containers are neglected;
- CPLEX is used for solving the optimization problem. We set the maximal computation time as 10 hours;
- Between the interval of the planning for each job, we set the time limit as 5 minutes for CPLEX to continue searching for its optimal solution.

6.4.2 Results and discussion

In this section, we evaluate the results of the proposed collision-free scheduling. Table 6.5 presents the initial solution of the proposed collision-free scheduling. Table 6.5 compares the makespan of the hybrid flow shop scheduling without considering collision avoidance and the collision-free scheduling taking collision avoidance into account. Table 6.7 provides the main computational performance for generating the collision-free trajectory planning of AGVs using the proposed approach. Table 6.8 gives the average operation time in comparison to the hybrid flow shop scheduling. For illustrating the procedure of generating trajectories of AGVs, we use the case of 3QC-1 as shown in Fig. 6.9. Fig. 6.10-Fig. 6.12 highlight the safety of AGVs are satisfied. Fig. 6.13 compares the proposed approach on the average distance of AGVs with mesh routing.

Table 6.5 presents the results of solving the hybrid flow shop scheduling problem for all experiments. The shown schedule solves a three-stage hybrid flow shop problem without the consideration of collision avoidance. In the case of 3QCs, the optimality of the optimization problem is achieved. However, in the case of 4QCs and 5QCs, the optimality of the optimization problem is not achieved due to its strong NP-hardness.

Table 6.6: The comparison of the makespan with and without collision-free trajectory.

Case	hybrid flow shop scheduling	collision-free scheduling
3QC-1	412s	412s
3QC-2	412s	412s
3QC-3	412s	412s
3QC-4	412s	412s
3QC-5	412s	414s
4QC-1	423s	413s
4QC-2	429s	429s
4QC-3	418s	413s
4QC-4	427s	420s
4QC-5	452s	419s
5QC-1	450s	432s
5QC-2	475s	472s
5QC-3	430s	416s
5QC-4	428s	428s
5QC-5	420s	420s

Table 6.6 compares the makespan of hybrid flow shop scheduling of the initial solution and makespan of the collision-free scheduling using the sequential planning approach. It is observed for the case of 3QCs the makespan of the collision-free scheduling is no less than the initial solution, considering the initial solution it is optimal. When it comes to the case of 4QCs and 5QCs, since the initial solutions are not optimal, the solver is still searching for the optimal solution during the sequential planning of all jobs.

Table 6.7 shows the main computational performance for generating the collision-free trajectory planning of AGVs using the proposed sequential approach. The planning of collision-free trajectories for the operation of AGVs can be computed in short time by solving a collection of MILPs. For a particular job to be planned, the variables in the hybrid flow shop can be updated quickly during the transition between O_i^{21} and O_i^{22} because it only involves a linear programming problem.

Table 6.8 compares the proposed approach regarding the average operation time processed by AGVs with the result of the hybrid flow shop scheduling. The average operation time processed by AGVs. It is noticed that the average operation times of the proposed approach are longer than the ideal case. This is because AGVs have to stay away from each other for avoiding collision, leading to longer average operation times for all AGVs.

Fig. 6.9 illustrates the partial procedure of generating trajectories of employed AGVs job by job in the case of 3QC-1. In the case of 3QC-1, the overall graph sequence is obtained as $\{18,2,15,17,13,6,21,8,14,20,5,12,23,4,11,24,16,7,19,1,9,22,10,3\}$. Using this overall graph sequence, the time window incorporating the collision-free trajectory for each job is generated sequentially, as shown in the sub-figures of Fig. 6.9(a)-6.9(h).

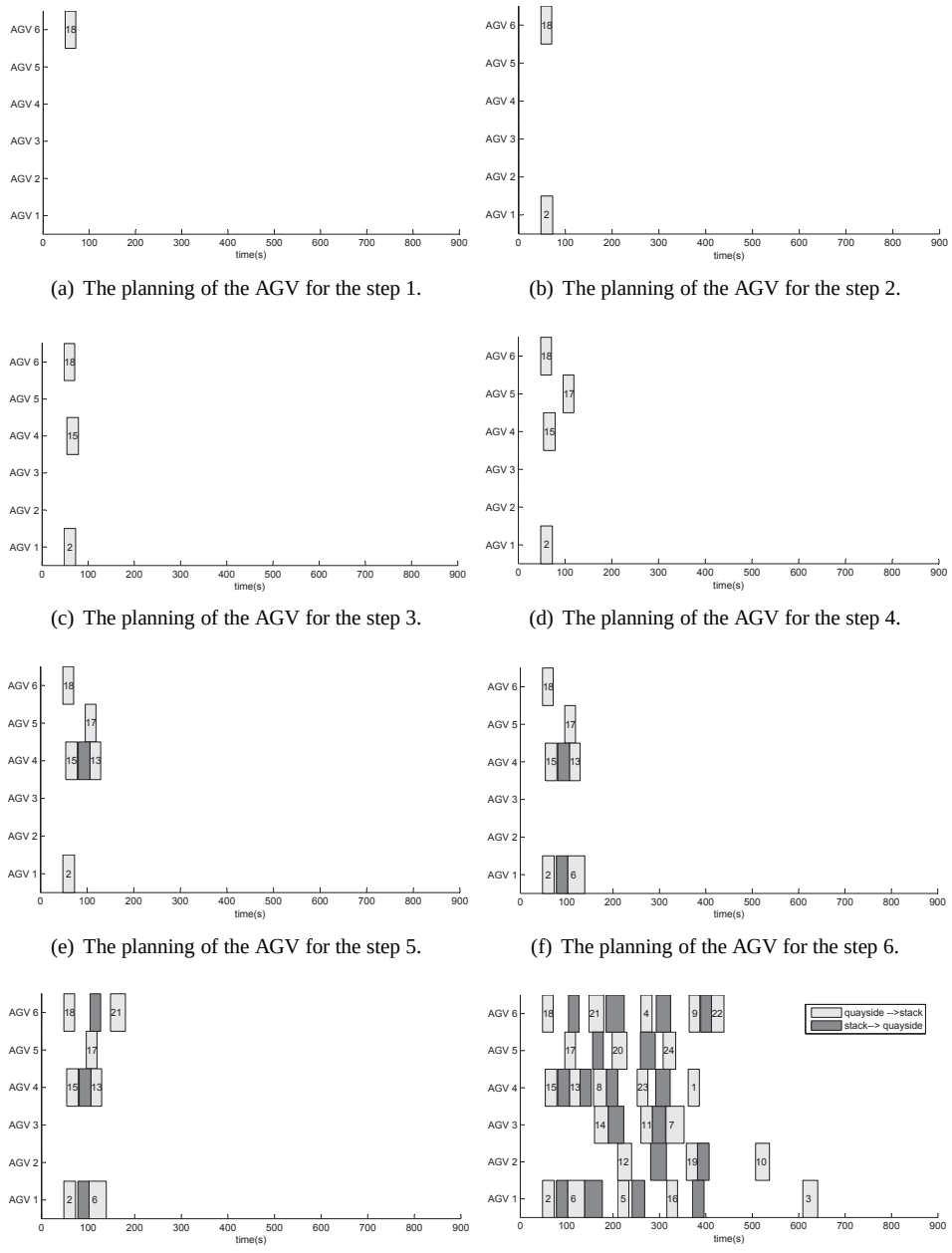
Fig. 6.10-6.12 present the relative distance for all paired AGVs in the case of 3 QCs, 4 QCs and 5 QCs, respectively. The trajectory for the operation processed by a particular AGV is planned for avoiding the potential collision of other AGVs. This shows that the

Table 6.7: The computation time of collision-free planning.

Case	average computation time for trajectory planning	average computation time for updating variables
3QC-1	13.46s	0.14s
3QC-2	14.56s	0.19s
3QC-3	15.27s	0.13s
3QC-4	48.53s	0.28s
3QC-5	10.33s	0.13s
4QC-1	9.93s	0.20s
4QC-2	64.19s	0.22s
4QC-3	9.94s	0.19s
4QC-4	9.42s	0.19s
4QC-5	9.43s	0.19s
5QC-1	11.27s	0.36s
5QC-2	14.45s	0.59s
5QC-3	13.73s	0.42s
5QC-4	12.83s	0.36s
5QC-5	11.79s	0.37s

Table 6.8: The comparison of the average operation times by AGVs. (unit seconds)

Case	hybrid flow shop	collision-free scheduling
3QC-1	26.74	27.76
3QC-2	27.62	28.47
3QC-3	28.10	29.19
3QC-4	27.29	28.07
3QC-5	26.38	27.45
4QC-1	25.80	26.71
4QC-2	25.91	27.28
4QC-3	26.23	26.70
4QC-4	25.88	26.91
4QC-5	25.90	26.79
5QC-1	26.60	27.77
5QC-2	27.76	28.59
5QC-3	26.26	27.94
5QC-4	27.01	28.10
5QC-5	27.80	28.58



(a) The planning of the AGV for the step 1.

(b) The planning of the AGV for the step 2.

(c) The planning of the AGV for the step 3.

(d) The planning of the AGV for the step 4.

(e) The planning of the AGV for the step 5.

(f) The planning of the AGV for the step 6.

(g) The planning of the AGV for the step 7.

(h) The overall planning of employed AGVs.

Figure 6.9: The illustration of the trajectory planning for employed AGVs in the case of 3QC-1.

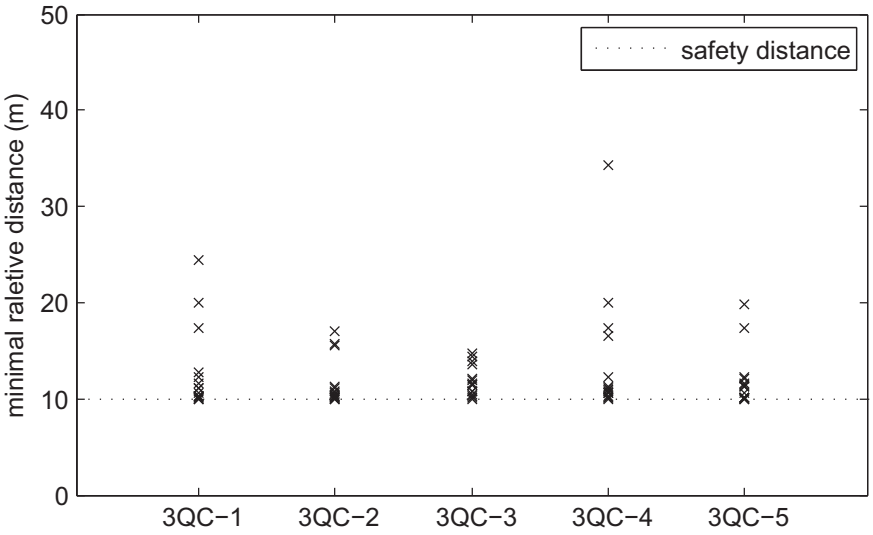


Figure 6.10: The relative distances of AGVs in the case of 3QCs.

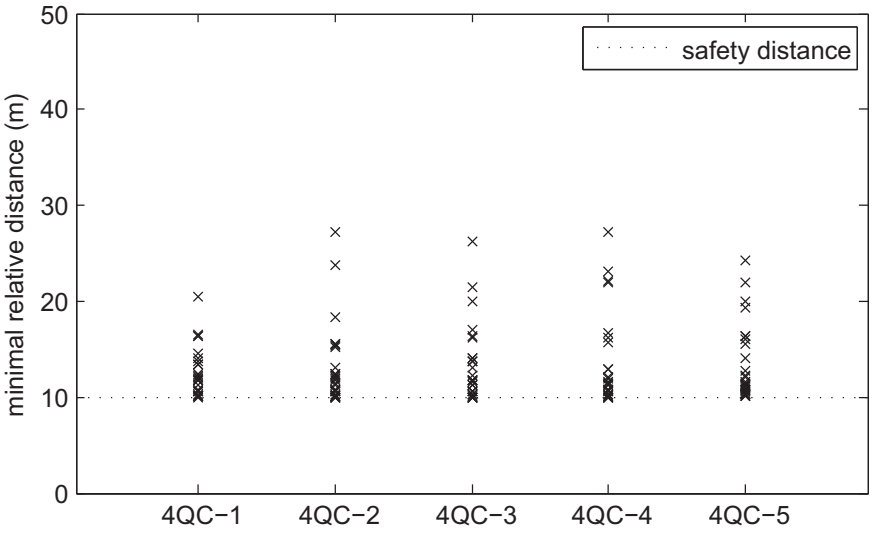


Figure 6.11: The relative distances of AGVs in the case of 4QCs.

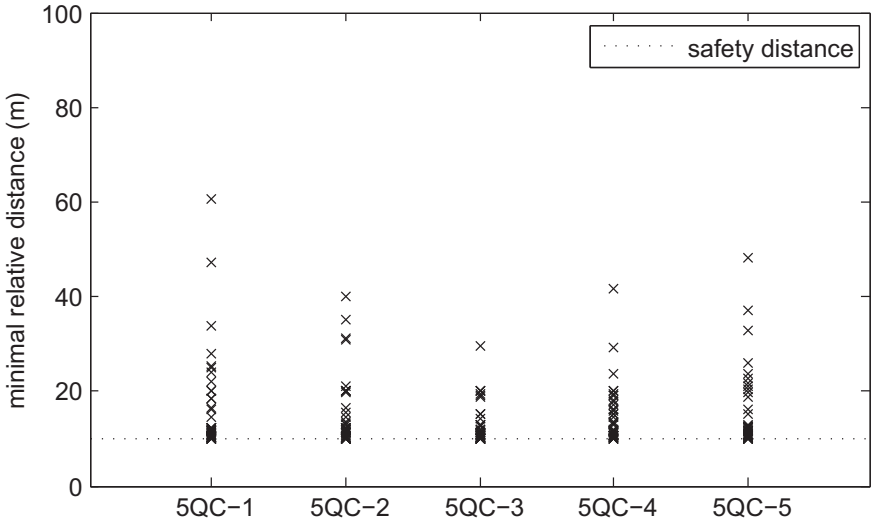


Figure 6.12: The relative distances of AGVs in the case of 5QCs.

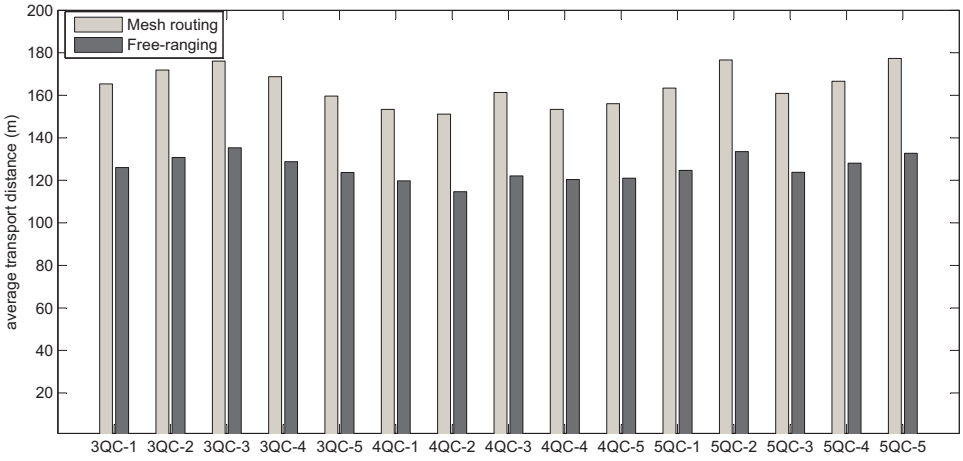


Figure 6.13: The comparison of the average transport distance of AGVs.

relative distance of any two AGVs is no less than 10 meters in all simulations.

Fig. 6.13 compares the average distance of the operations processed by AGVs. In this comparison, the proposed approach and the mesh routing are considered. The mesh routing is typically used for AGVs in automated container terminals [30]. This figure shows that the proposed approach can reduce the average distance of AGVs more than 20 % compared to mesh routing. The reduction of the average distance can be found for each experiment consistently.

6.5 Concluding remarks

This chapter investigates the collision-free scheduling of interacting machines in a large-size container terminal, which involves multiple QCs, multiple AGVs and multiple ASCs. The problem combines discrete-event dynamics and continuous-time dynamics of three types of machines. In particular, for safety considerations the static obstacles and the dynamical obstacles of AGVs are modeled explicitly in this chapter. Based on a hierarchical control architecture for coordinating discrete-event dynamics and continuous-time dynamics, we propose an approach for generating sequentially collision-free trajectories of AGVs.

The performance of the proposed algorithm is tested in the simulations of a large-size terminal. Simulation results illustrate that the proposed approach is able to ensure no collisions. It is also observed that this is at the cost of a slightly longer makespan. The proposed approach can reduce the average distance of AGV operations significantly, compared with the conventional mesh routing currently typically used in automated container terminals.

Chapter 7

Conclusions and recommendations

This thesis focuses on operational control of automated container terminals, wherein the energy efficiency and the implementation of free-ranging AGVs have been particularly investigated. As a concluding chapter, this final chapter first presents the main conclusions of this thesis and answers the related research questions. Then remaining open problems that are recommended in future research are presented.

7.1 Conclusions

In this thesis, the following main research question is addressed: *how to improve energy efficiency and implement autonomously moving equipment of automated container terminals at the operational level?* In order to investigate this, for automated container terminals, this thesis considers three types of terminals: the compact terminal, the medium-size terminal and the large-size terminal. Chapter 3, 4 and 5 indeed address the problem of improving the energy efficiency of the compact terminal and the medium-size terminal. Chapter 6 investigates the implementation of free-ranging AGVs in the large-size terminal.

Following the main research question, the three sub-questions formulated in Chapter 1 are answered as follows:

- *To what extent can the energy consumption be reduced while maintaining an acceptable operational performance?*

  The energy efficiency has been discussed in Chapter 3-5 for the compact terminal and the medium-size terminal. For the compact terminal, on average 27 % of energy consumption for horizontal transport is saved for the minimal completion time. When it comes to the medium-size container terminal, the average energy consumption for horizontal transport is reduced by 37 % for the minimal completion time.

- *What complexity of control algorithms should be considered?*

  For the compact terminal, the discrete-event dynamics and the continuous-time dynamics are controlled by a centralized controller as a whole. Using the centralized

way, a hybrid model predictive control is proposed for achieving energy efficiency for real-time operations by controlling each piece of equipment directly. The optimization problem is formulated as a mixed integer linear programming problem that can be efficiently solved by commercial solvers (e.g., CPLEX).

For the medium-size and the large-size terminal, as the system scale increases, more pieces of equipment are considered. Compared to the compact terminal, a large number of decision variables, in particular discrete variables for determining the job sequence that is processed by a particular piece of equipment, have to be considered additionally. Therefore, the complexity for controlling these pieces of equipment grows significantly. To reduce the control complexity of the medium-size and the large-size terminal, a hierarchical control architecture is proposed for decoupling the hybrid dynamical system. In this hierarchical architecture, a higher-level controller is responsible for the discrete-event dynamics while the lower-level controller is concerned with the continuous-time dynamics for each piece of equipment.

For the medium-size terminal, the coordination between the higher-level controller and the lower-level controller is considered for achieving energy efficiency. For the open loop control problem (Chapter 4), the coordination between these two levels is considered once beforehand. For the closed-loop control problem (Chapter 5), the coordination between these two levels is considered multiple times, each of which is carried out when a particular even takes place for rescheduling.

The large-size terminal, which is the most complex case, involves collision-free trajectory planning of free-ranging AGVs integrated with the scheduling of interacting machines. For this complex overall problem, a hierarchical control architecture still used for coordinating different levels of controllers. Considering the collision avoidance, additional coordination between different controllers of AGVs are considered. Using this extended hierarchical control architecture, a sequential planning approach is proposed using multiple coordinations between different levels and within the lower level.

- *How can the collision-free trajectory planning of AGVs and other equipment be integrated with the scheduling of interacting machines in automated container terminals?*

  In this overall problem, the collision-free trajectory planning of free-ranging AGVs is coupled with scheduling of interacting machines. In Chapter 6, a sequential planning approach is proposed for addressing this research question. This sequential planning approach uses a hierarchical architecture for coordinating the controllers at different levels. The detailed schedule is determined job by job following a particular overall graph sequence. For a particular job, the subproblems include a hybrid flow shop scheduling problem at the high-level controller and one or two small-scale mixed integer linear programming problem at the lower-level controller for determining the collision-free trajectory of a free-ranging AGV.

7.2 Recommendations for future research

This thesis uses the perspective of hybrid systems for the operational control of automated container terminals, providing some new results for improving the terminal performance.

Still, further investigations needs to be considered before these results can be used for practical operations. Besides this, inspired by this thesis, additional directions are also recommended.

7.2.1 Operational control of container terminals

This thesis focuses on the energy efficiency and implementation of free-ranging AGVs of automated container terminals. However, the operations of container terminals are considerably dynamic and complex, involving a great number of decisions for the terminal control. The implications resulting from practical operations motivate further investigations, which are provided as follows:

- Modeling accuracy

  This thesis considers a double integrator for the continuous-time dynamics of a piece of equipment. For future research, a nonlinear model including the air-drag, the rolling resistance, etc., will be considered, resulting in a more accurate model of the continuous-time dynamics. Due to this nonlinear dynamical model, the energy consumption of the piece of equipment for transporting containers needs to be adjusted as well. For the energy efficiency, the overall problem involves the minimization of the completion time of the hybrid flow shop scheduling and the minimization of the energy consumption of the nonlinear continuous-time dynamics of the piece of equipment.

  For the collision-free trajectory planning of free-ranging AGVs, the heading of the AGV needs to be modeled due to the constraint on the heading at the transport point. Besides this, a more exact representation of the dynamical collision area of AGVs needs to be investigated, which reduces the occupancy area of AGVs in the quayside transport area and allows other vehicles to move with more space.

- Operation complexity

  This thesis focuses on the horizontal operation of the pieces of equipment employed for transporting containers. Regarding the scheduling of interacting machines, the typical scenario considered in this thesis is unloading a vessel, in which a number of containers are located in a single layer in a particular bay. For a more practical situation, simultaneous loading and unloading of a large number of containers of the vessel, in which additional position constrains for multiple layers of containers must be considered, will be included in future research.

  For the collision-free scheduling of free-ranging AGVs, in this thesis the capacity of the vehicle and the capacity of the transfer point are both limited. For the practical situation, the capacity of two AGVs for 2 TEUs and multiple transport points of the stacking area needs to be considered in future research.

- Operational uncertainties

  For the compact and the medium-size terminal, this thesis considers two types of operational uncertainties: the operational delay and the precise arrival time of new containers. Besides, other possible uncertainties (e.g. the breakdown of a piece of

equipment) needs to be considered in future research, resulting in a more robust control algorithm.

For the large-size terminal, the only uncertainty considered in this thesis is the possible collision avoidance between different AGVs. For future research, external uncertainties (e.g., the operational delay and the breakdown of the piece of equipment) will be considered for real-time operational control.

- Computational efficiency

  Both for energy efficiency and implementing free-ranging AGVs, the hybrid flow shop scheduling related to the discrete-event dynamics is considered in the higher-level controller of the proposed hierarchical architecture. This thesis focuses on the interaction of the discrete-event dynamics and the continuous-time dynamics. The hybrid flow shop scheduling is known to be NP-hard. The computational efficiency of the hybrid flow shop scheduling problem needs to be investigated for a fast acceptable solution (e.g. using meta heuristics), which is expected for terminal operators.

7.2.2 Additional directions for future research

Although the operational control of automated container terminals has its particular features with respect to containers, it has close connections with other types of material handling (e.g., dry bulk). Therefore, the directions of these research problems can also be recommended additionally after investigating automated container terminals.

- Operational control of dry bulk terminals

  Besides container transport, dry bulk transport (e.g., iron ore and coal) also plays an important role in intermodal transport. The operational control of a dry bulk terminal is still an open problem for investigation. The dynamics of a dry bulk terminal include continuous-time dynamics (e.g., a flow of material using belt conveyors) and discrete-event dynamics (e.g., the stack option of a stack reclaimer). Therefore, using a flow perspective, hybrid model predictive control can be considered as a promising technology for the operational control of a dry bulk terminal, which minimizes the cost of related operations.

Bibliography

[1] M. Acciaro, H. Ghiara, and M.I. Cusano. Energy management in seaports: A new role for port authorities. *Energy Policy*, 71:4–12, 2014.

[2] A. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero, and G. Soncin. Modeling and feedback control for resource allocation and performance analysis in container terminals. *IEEE Transactions on Intelligent Transportation Systems*, 9(4):601–614, 2008.

[3] P. Angeloudis and M.G.H. Bell. An uncertainty-aware AGV assignment algorithm for automated container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 46(3):354–356, 2010.

[4] A.N.Venkat, I.A. Hiskens, J.B.Rawlings, and S.J. Wright. Distributed mpc strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6):1192–1206, 2008.

[5] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(1):407–427, 1999.

[6] L. Ben-Naoum, R. Boel, L. Bongaerts, B. De Schutter, Y. Peng, P. Valckenaers, J. Vandewalle, and V. Wertz. Methodologies for discrete event dynamic systems: A survey. *Journal A*, 36(4):3–14, 1995.

[7] M. Bielli, A. Boulmakoul, and M. Rida. Object oriented model for container terminal distributed simulation. *European Journal of Operational Research*, 175(3):1731–1751, 2006.

[8] C. Bierwirth and F. Meisel. A fast heuristic for quay crane scheduling with interference constraints. *Journal of Scheduling*, 12(4):345–360, 2009.

[9] C. Bierwirth and F. Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3): 615–627, 2010.

[10] D. Briskorn, A. Drexl, and S. Hartmann. Inventory-based dispatching of automated guided vehicles on container terminals. *OR Spectrum*, 28(4):611–630, 2006.

[11] B. Cai, S. Huang, D. Liu, and G. Dissanayake. Rescheduling policies for large-scale task allocation of autonomous straddle carriers under uncertainty at automated container terminals. *Robotics and Autonomous Systems*, 62(4):506–514, 2014.

[12] E.F. Camacho, D.R. Ramírez, D. Limón, D. Muñoz de la Peña, and T. Alamo. Model predictive control techniques for hybrid systems. *Annual reviews in control*, 34(1): 21–31, 2010.

[13] E. Camponogara, D. Jia, B.H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1):44–52, 2002.

[14] J. X. Cao, D. Lee, J. H. Chen, and Q. Shi. The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. *Transportation Research Part E: Logistics and Transportation Review*, 46(3):344–353, 2010.

[15] H.J. Carlo, I.F. Vis, and K.J. Roodbergen. Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research*, 236(1):1–13, 2014.

[16] L. Chen and A. Langevin. Multiple yard cranes scheduling for loading operations in a container terminal. *Engineering Optimization*, 43(11):1205–1221, 2011.

[17] L. Chen, N. Bostel, P. Dejax, J. Cai, and L Xi. A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *European Journal of Operational Research*, 181(1):40–58, 2007.

[18] L. Chen, A. Langevin, and Z. Lu. Integrated scheduling of crane handling and truck transportation in a maritime container terminal. *European Journal of Operational Research*, 225(1):142–152, 2013.

[19] Clarkson. Clarkson research services limited. URL: www.crsl.com, Janurary 2015.

[20] T.G. Crainic and K.H. Kim. Intermodal transportation. In C. Barnhart and G. Laporte, editors, *Handbooks in Operations Research and Management Science*, volume 14 of *Transporation*, pages 467–536. Elsevier, Amsterdam, The Netherlands, 2007.

[21] J. Currie and D.I. Wilson. OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User. In N. Sahinidis and J. Pinto, editors, *Foundations of Computer-Aided Process Operations*, Savannah, Georgia, 2012.

[22] K.H. Van Dam, Z. Lukszo, J.A. Ottjes, and G. Lodewijks. Distributed intelligence in autonomous multi-vehicle systems. *International journal of critical infrastructures*, 2(2):261–272, 2006.

[23] R. De Koster, B. Balk, and W. Van Nus. On using DEA for benchmarking container terminals. *International Journal of Operations & Production Management*, 29(11): 1140–1155, 2009.

[24] L.B. de Oliveira and E. Camponogara. Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(1):120–139, 2010.

[25] R. Dekker, J. Bloemhof, and I. Mallidis. Operations research for green logistics–an overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, 219(3):671–679, 2012.

[26] A. Diabat and E. Theodorou. An integrated quay crane assignment and scheduling problem. *Computers & Industrial Engineering*, 73:115–123, 2014.

[27] M.D. Doan, P. Giselsson, T. Keviczky, B. De Schutter, and A. Rantzer. A distributed accelerated gradient algorithm for distributed model predictive control of a hydro power valley. *Control Engineering Practice*, 21(11):1594–1605, 2013.

[28] Drewry. Drewry shipping consultants ltd. URL: www.drewry.co.uk, Janurary 2015.

[29] M. B. Duinkerken and J. A. Ottjes. A simulation model for automated container terminals. In *Proceedings of the Business and Industry Simulation Symposium*, volume 10, pages 134–139, Washington D.C., April 2000.

[30] M. B. Duinkerken, M. Van der Zee, and G. Lodewijks. Dynamic free range routing for automated guided vehicles. In *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control*, pages 312–317, Ft. Lauderdale, Florida, 2006.

[31] M.B. Duinkerken, J.A. Ottjes, and G. Lodewijks. Comparison of routing strategies for agv systems using simulation. In *Proceedings of the 38th conference on Winter simulation*, pages 1523–1530, Monterey, California, 2006.

[32] ECT. ECT Delta Terminal. URL: http://myservices.ect.nl/Terminals/rotterdamterminals/deltaterminal, 2015. Last accessed on 1st Janurary 2015.

[33] K. Edlund, J.D. Bendtsen, and J.B. Jørgensen. Hierarchical model-based predictive control of a power plant portfolio. *Control Engineering Practice*, 19(10):1126–1136, 2011.

[34] E. Gawrilow, E. Köhler, R.H. Möhring, and B. Stenzel. Dynamic routing of automated guided vehicles in real-time. In *Mathematics–Key Technology for the Future*, pages 165–177. Springer, 2008.

[35] Hans P. Geering. *Optimal Control with Engineering Applications*. Springer, Berlin, Germany, 2007.

[36] A.H. Gharehgozli, G. Laporte, Y. Yu, and R. de Koster. Scheduling twin yard cranes in a container block. *Transportation Science*, 2014.

[37] A.H. Gharehgozli, Y. Yu, R. de Koster, and J.T. Udding. An exact method for scheduling a yard crane. *European Journal of Operational Research*, 235(2):431–447, 2014.

[38] 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.

[39] D.N. Godbole, J. Lygeros, and S. Sastry. Hierarchical hybrid control: A case study. In *Proceedings of the 34th IEEE Conference on Decision and Control*, Louisiana, New Orleans, 1994.

[40] K. Gokbayrak and C.G. Cassandras. A hierarchical decomposition method for optimal control of hybrid systems. In *Proceedings of the 16th International IEEE Conference on Decision and Control*, pages 1816–1921, Sydney, Australia, 2000.

[41] Gottwald. Gottwald AGV Automated Guided Vehicles. URL:http://www.gottwald.com/, 2012. Last accessed on 1st September 2012.

[42] Gottwald. Gottwald ASC Automated Stacking Cranes. URL: http://www.gottwald.com/, 2012. Last accessed on 1st September 2012.

[43] M. Grunow, H. Günther, and M. Lehmann. Dispatching multi-load agvs in highly automated seaport container terminals. *OR Spectrum*, 26(2):211–235, 2004.

[44] J.M. Guerrero, J.C. Vasquez, J. Matas, L.G. de Vicuña, and M. Castilla. Hierarchical control of droop-controlled AC and DC microgrids: a general approach toward standardization. *IEEE Transactions on Industrial Electronics*, 58(1):158–172, 2011.

[45] B. Ha, E. Park, and C. Lee. A simulation model with a low level of detail for container terminals and its applications. In *Proceedings of the 2007 Winter Simulation Conference*, pages 2003–2011, Washington D.C., December 2007.

[46] J. He, D. Chang, W. Mi, and W. Yan. A hybrid parallel genetic algorithm for yard crane scheduling. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):136–155, 2010.

[47] M.P.M Hendriks. *Multi-step optimization of logistics networks: strategic, tactical, and operational decisions*. PhD thesis, Eindhoven University of Technology, 2009.

[48] L. Henesey. *Enhancing Container Terminal Performance A Multi Agent Systems Approach*. PhD thesis, Blekinge Institute of Technology, 2004.

[49] L. Henesey, P. Davidsson, and J. Persson. Agent based simulation architecture for evaluating operational policies in transshipping containers. *Autonomous Agents and Multi-Agent Systems*, 18(2):220–238, 2009.

[50] T. Henzinger. The theory of hybrid automata. In *Proceedings of of the 11th IEEE Symposium on Logic in Computer Science*, pages 278–292, New Brunswick, New Jersey, 1996.

[51] M. Houwing, R.R. Negenborn, and B. De Schutter. Demand response with micro-chp systems. *Proceedings of the IEEE*, 99(1):200–213, 2011.

[52] IHS. Ihs global insight. URL: www.ihsglobalinsight.com, Janurary 2015.

[53] P.A. Ioannou. *Intelligent Freight Transportation*. CRC press, Parkway, NW, 2014.

[54] Jurong Port. Doosan Quay Cranes. URL: http://www.jp.com.sg/JurongPort, 2011. Last accessed on 1st September 2012.

[55] I. Kamwa, R. Grondin, and Y. Hebert. Wide-area measurement based stabilizing control of large power systems-a decentralized/hierarchical approach. *IEEE Transactions on Power Systems*, 16(1):136–153, 2001.

[56] T. Keviczky, F. Borrelli, and G.J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, 2006.

[57] 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.

[58] K.H. Kim, S.M. Jeon, and K.R. Ryu. Deadlock prevention for automated guided vehicles in automated container terminals. *OR Spectrum*, 28(4):659–679, 2006.

[59] K.P. Kim and J.W. Bae. A look-ahead dispatching method for automated guided vehicles in automated port container terminals. *Transportation science*, 38(2):224–234, 2004.

[60] K.P. Kim and Y.M. Park. A crane scheduling method for port container terminals. *European Journal of operational research*, 156(3):752–768, 2004.

[61] Y. Kuwata, A. Richards, T. Schouwenaars, and J.P. How. Distributed robust receding horizon control for multivehicle guidance. *IEEE Transactions on Control Systems Technology*, 15(4):627–641, 2007.

[62] H.Y.K. Lau and Y. Zhao. Integrated scheduling of handling equipment at automated container terminals. *International journal of production economics*, 112(2):665–682, 2008.

[63] J.H. Leeper. Integrated automated terminal operations. *Transportation Research Circular*, 33:23–28, 1988.

[64] C. Liang, Y. Huang, and Y. Yang. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers and Industrial Engineering*, 56(3):1021–1028, 2009.

[65] C.I. Liu, H. Jula, and P. Ioannou. Design, simulation, and evaluation of automated container terminals. *IEEE Transactions on Intelligent Transportation Systems*, 3(1): 12–26, 2002.

[66] C.I. Liu, H. Jula, K. Vukadinovic, and P. Ioannou. Automated guided vehicle system for two container yard layouts. *Transportation Research Part C: Emerging Technologies*, 12(5):349–368, 2004.

[67] J. Lunze. *Feedback control of large scale systems*. Prentice Hall, London, UK, 1992.

[68] Michael M. Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, West Sussex, England, 2009.

[69] J.M. Maestre and R.R. Negenborn. *Distributed model predictive control made easy*. Springer, Dordrecht, The Netherlands, 2014.

[70] J.M. Maestre, D. Muñoz de la Peña, E.F. Camacho, and T. Alamo. Distributed model predictive control based on agent negotiation. *Journal of Process Control*, 21(5): 685–697, 2011.

[71] J. Martin and J.J. Odel. *Object-Oriented Analysis and Design*. Prentice-Hall, Upper Saddle River, NJ, 1992.

[72] P.J.M. Meersmans and A.P.M. Wagelmans. Effective algorithms for integrated scheduling of handling equipment at automated container terminals. Technical report, Econometric Institute, Erasmus University Rotterdam, Rotterdam, The Netherlands, 2001.

[73] J.L. Nabais, R.R. Negenborn, R.B. Carmona-Benítez, L.F. Mendonça, and B.A. Botto. Hierarchical MPC for multiple commodity transportation networks. In *Distributed Model Predictive Control Made Easy*, pages 535–552. Springer, Dordrecht, The netherlands, 2014.

[74] R.R. Negenborn, P.van Overloop, T. Keviczky, and B. De Schutter. Distributed model predictive control of irrigation canals. *Networks and Heterogeneous Media*, 4(2): 359–380, 2009.

[75] W.C. Ng and K.L. Mak. Yard crane scheduling in port container terminals. *Applied Mathematical Modelling*, 29(1):263–276, 2006.

[76] C. Ocampo-Martínez, S. Bovo, and V. Puig. Partitioning approach oriented to the decentralised predictive control of large-scale systems. *Journal of Process Control*, 21(5):775–786, 2011.

[77] C. Ocampo-Martínez, D. Barcelli, V. Puig, and A. Bemporad. Hierarchical and decentralised model predictive control of drinking water networks: application to barcelona case study. *IET Control Theory & Applications*, 6(1):62–71, 2012.

[78] D. Ouelhadj and S. Petrovic. A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4):417–431, 2009.

[79] I. Papamichail, A. Kotsialos, I.Margonis, and M.Papageorgiou. Coordinated ramp metering for freeway networks–a model-predictive hierarchical control approach. *Transportation Research Part C: Emerging Technologies*, 18(3):311–331, 2010.

[80] Robbert R. van Zijverden and R.R. Negenborn. Survey of approaches for integrated control of intermodal container terminals. In *2012 9th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 67–72, 2012.

[81] P.J.G. Ramadge and W.M. Wonham. The control of discrete event dynamical systems. In *Proceedings of IEEE*, volume 77, pages 81–98, 1989.

[82] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin, 2009.

[83] A. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the 2002 American Control Conference*, volume 3, pages 1936–1941, Anchorage, Alaska, 2002.

[84] J.C. Rijsenbrij and A. Wieschemann. Sustainable container terminals: a design approach. In *Handbook of terminal planning*, pages 61–82. Springer, Heidelberg, Germany, 2011.

[85] J.P. Rodrigue, C. Comtois, and B. Slack. *The Geography of Transport Systems*. Routledge, New York, NY, 2013.

[86] R. Ruiz and J.A. Vázquez-Rodríguez. The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1):1–18, 2010.

[87] R. Scattolini. Architectures for distributed and hierarchical model predictive control–a review. *Journal of Process Control*, 19(5):723–731, 2009.

[88] T. Schouwenaars, B.D. Moor, E. Feron, and J. How. Mixed integer programming for multi-vehicle path planning. In *Proceedings of the European Control Conference*, pages 2603–2608, Porto, Portugal, 2001.

[89] B. D. Schutter and M. Heemels. *Lecture notes for modeling and control of hybrid systems*. Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, 2001.

[90] A. Sciomachen and E. Tanfani. A 3d-bpp approach for optimising stowage plans and terminal productivity. *European Journal of Operational Research*, 183(3):1433–1446, 2007.

[91] M.J. Sharma and S.J. Yu. Benchmark optimization and attribute identification for improvement of container terminals. *European Journal of Operational Research*, 201(2):568–580, 2010.

[92] D.D. Siljak. *Decentralized control of complex systems*. Academic Press, Boston, Massachusetts, 2011.

[93] R. Stahlbock and S. Voß. Operations research at container terminals: a literature update. *OR Spectrum*, 30(1):1–52, 2007.

[94] D. Steenken, S. Voß, and R. Stahlbock. Container terminal operation and operations research: A classification and literature review. *OR Spectrum*, 26(1):3–49, 2004.

[95] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460–469, 2010.

[96] TBA. TBA simulation. URL: http://www.tba.nl, March 2013.

[97] F.D. Torrisi and A. Bemporad. Hysdel–a tool for generating computational hybrid models for analysis and synthesis problems. *IEEE Transactions on Control Systems Technology*, 12(2):235–249, 2004.

[98] T.J.J. van den Boom and B. De Schutter. Modelling and control of discrete event systems using switching max-plus-linear systems. *Control engineering practice*, 14 (10):1199–1211, 2006.

[99] J. H. R van Duin and H. Geerlings. Estimating CO2 footprints of container terminal port-operations. *International Journal of Sustainable Development and Planning*, 6 (4):459–473, 2011.

[100] I.F.A. Vis. Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170(3):677–709, 2006.

[101] I.F.A. Vis and R. de Koster. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147(1):1–16, 2003.

[102] I.F.A. Vis and I. Harika. Comparison of vehicle types at an automated container terminal. *OR Spectrum*, 26(1):117–143, 2004.

[103] I.F.A. Vis, R. de Koster, K.J. Roodbergen, and L.W.P Peeters. Determination of the number of automated guided vehicles required at a semi-automated container terminal. *Journal of the Operational Research Society*, 52(4):409–417, 2001.

[104] C. Wang and C.J. Ong. Distributed model predictive control of dynamically decoupled systems with coupled cost. *Automatica*, 46(12):2053–2058, 2010.

[105] E.T.J. Wijlhuizen and F. Meeussen. The sustainable container terminal (Maasvlakte II). In *Proceedings of the 2nd International Conference on Harbours, Air Quality and Climate Change*, Rotterdam, The Netherlands, 2008.

[106] F.Y. Xiao, P.K. Li, and H.C. Chun. A distributed agent system for port planning and scheduling. *Advanced Engineering Informatics*, 25(3):403–412, 2011.

[107] J. Xin, R. R. Negenborn, and G. Lodewijks. Hybrid model predictive control for equipment of an automated container terminal. In *Proceedings of the 2013 IEEE International Conference on Networking, Sensing and Control*, pages 746–752, Evry, France, 2013.

[108] J. Xin, R.R. Negenborn, and G. Lodewijks. Hybrid MPC for balancing throughput and energy consumption in an automated container terminal. In *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems*, pages 1238–1244, The Hague, The Netherlands, 2013.

[109] J. Xin, R.R. Negenborn, and G. Lodewijks. Hierarchical control of equipment in automated container terminals. In *Proceedings of the 4th International Conference on Computational Logistics*, pages 1–17, Copenhagen, Denmark, September 2013.

[110] 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.

[111] J. Xin, R.R. Negenborn, and G. Lodewijks. Rescheduling of interacting machines in automated container terminals. In *Proceedings of the 19th IFAC World Congress*, pages 1698–1704, Cape Town, South Africa, 2014.

[112] J. Xin, R.R. Negenborn, and G. Lodewijks. Trajectory planning for agvs in automated container terminals using avoidance constraints. In *Proceedings of the 19th IFAC World Congress*, pages 9828–9833, Cape Town, South Africa, 2014.

[113] 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. Technical report, Delft Univeristy of Technology, Janurary 2015. Submitted to a journal.

[114] J. Xin, R.R. Negenborn, and G. Lodewijks. Event-driven receding horizon control for energy-efficient container handling. *Control Engineering Practice*, 39:45–55, 2015.

[115] J. Xin, R.R. Negenborn, and G. Lodewijks. Energy-efficient container handling using hybrid model predictive control. *International Journal of Control*, 2015. Accpted for publication.

[116] W.Y. Yun and Y.S. Choi. A simulation model for container-terminal operation analysis using an object-oriented approach. *International Journal of Production Economics*, 59(1):221–230, 1999.

[117] A. Zafra-Cabeza, J.M. Maestre, M.A. Ridao, E.F. Camacho, and L. Sánchez. A hierarchical distributed model predictive control approach to irrigation canals: A risk mitigation perspective. *Journal of Process Control*, 21(5):787–799, 2011.

[118] J. Zeng and W.J. Hsu. Conflict-free container routing in mesh yard layouts. *Robotics and Autonomous Systems*, 56(5):451–460, 2008.

[119] H. Zhang and K.H. Kim. Maximizing the number of dual-cycle operations of quay cranes in container terminals. *Computers & Industrial Engineering*, 56(3):979–992, 2009.

[120] W. Zhang, M. Kamgarpour, D. Sun, and C.J. Tomlin. A hierarchical flight planning framework for air traffic management. *Proceedings of the IEEE*, 100(1):179–194, 2012.

[121] L. Zhen, L.H. Lee, E.P. Chew, D. Chang, and Z. Xu. A comparative study on two types of automated container terminal systems. *IEEE Transactions on Automation Science and Engineering*, 9(1):56–69, 2012.

Glossary

List of symbols and notations

Below follows a list of the most frequently used symbols and notations in this thesis.

a_i	the continuous variable associated with job i
$\mathbf{A}$	system matrices of linear time-invariant models
b_i	the continuous variable associated with job i
$b_{\text{in},\tau}$	the binary variable for modeling a static obstacle
$b_{\text{m},\tau}$	the binary variable for modeling a moving obstacle
$\mathbf{B}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$	input matrices of linear time-invariant models
c_i	the continuous variable associated with job i
$\mathbf{C}$	output matrices of linear time-invariant models
d_i	the continuous variable associated with job i
d_s	the safety distance
d_t	the traveling distance of a piece of equipment
$\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$	exogenous input matrices of linear time-invariant models
e_i	the continuous variable associated with job i
$\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \mathbf{E}_4$	real matrices of mixed-logical dynamic models
$\mathbf{E}_5$	a real vector of mixed-logical dynamic models
E_c	a finite set of edges of a controlled component
E_uc	a finite set of edges of an uncontrolled component
$E_\text{qc}, E_\text{agv}, E_\text{asc}$	energy consumption of the QC, the AGV and the ASC
E_tot	the total energy consumption of equipment
f_agv	the mapping function of the AGVs
f_asc	the mapping function of the ASCs
f_c	the continuous-time dynamics of a controlled component
f_qc	the mapping function of the QC(s)
f_uc	the continuous-time dynamics of an uncontrolled component
G_c	a finite set of guards of a controlled component
G_c^inter	a set of guards of a controlled component interacting with

	other hybrid systems
G_{uc}	a finite set of guards of an uncontrolled component
$G_{\text{uc}}^{\text{inter}}$	a set of guards of an uncontrolled component interacting with other hybrid systems
h_1	stage indices the hybrid flow shop
h_2	operation indicator for a particular stage in the hybrid flow shop
H	hybrid automaton
i	job indices
j	job indices
J	objective function
k	discretized time instant
k_{d}	the time instance at which a discrete event is triggered
l	indices of steps for MPC controller
$m_{\text{agv}}^{\text{load}}$	loading weight of the AGV with the container
$m_{\text{asc}}^{\text{load}}$	loading weight of the ASC with the container
$m_{\text{qc}}^{\text{load}}$	loading weight of the QC with the container
$m_{\text{agv}}^{\text{unload}}$	unloading weight of the AGV without the container
$m_{\text{asc}}^{\text{unload}}$	unloading weight of the ASC without the container
$m_{\text{qc}}^{\text{unload}}$	unloading weight of the QC without the container
M	a positive integer number used for approximation
n_{qc}	the number of the QCs
n_{agv}	the number of the AGVs
n_{asc}	the number of the ASCs
N	number of jobs
N_{p}	the prediction horizon of the MPC controller
N_{sim}	the simulation length
N_{uc}	the number of containers of an uncontrolled component
$\text{O}_i^{h_1 h_2}$	the operation associated with job i
p	the index of the AGV
P_i^1	the position of job i in the vessel related to Stage 1
P_i^2	the transfer point of job i near the quay crane related Stage 1 and 2
P_i^3	the transfer point of job i near the stack related Stage 2 and 3
P_i^4	the storage place of job i in the stack related to Stage 3
q_n	the element in Q
Q	the overall graph sequence
$\mathbf{r}$	the continuous-time state of a piece of equipment
$\mathbf{r}_0$	the initial continuous-time state of a particular operation

r_1	the position of a piece of equipment
r_2	the velocity of a piece of equipment
$\mathbf{r}_\mathrm{f}$	the final continuous-time state of a particular operation
R	a large positive number
R_c	a finite set of reset maps of a controlled component
R_uc	a finite set of reset maps of an uncontrolled component
$\mathbb{R}$	the set of real numbers
$s_i^{h_1h_2}$	the lower bound of the processing time of operation
$\mathrm{S_c}$	a finite set of discrete modes of a controlled component
$\mathrm{S_{uc}}$	the set of discrete modes of an uncontrolled component
$\|\mathbf{t}\|_1$	the sum of all operation times
t_f	the ending time of a particular operation
t_finish	the completion time
$t_i^{h_1h_2}$	the processing time of operation $\mathrm{O}_i^{h_1h_2}$
$t_{\mathrm{start},i}^{h_1h_2}$	the starting time of operation $\mathrm{O}_i^{h_1h_2}$
$t_{\mathrm{end},i}^{h_1h_2}$	the ending time of operation $\mathrm{O}_i^{h_1h_2}$
t_w	a number that ensures the calculation of the minimal time
t_0	the stating time of a particular operation
T_r	The remained time steps
u_c	the acceleration of a controlled component
$u_\mathrm{max}^\mathrm{agv}$	the maximal acceleration of the AGV
$u_\mathrm{max}^\mathrm{asc}$	the maximal acceleration of the ASC
$u_\mathrm{max}^\mathrm{qc}$	the maximal acceleration of the QC
u_max	the maximal acceleration of the equipment
u_min	the minimal acceleration of the equipment
$\mathbf{u}_p(k)$	the acceleration vector of the AGV at time instant k
u_uc	the acceleration of an uncontrolled component
U_c	a finite set of control variables of a controlled component
U_uc	a finite set of control variables of an uncontrolled component
$v_\mathrm{max}^\mathrm{agv}$	the maximal velocity of the AGV
$v_\mathrm{max}^\mathrm{asc}$	the maximal velocity of the ASC
$v_\mathrm{max}^\mathrm{qc}$	the maximal velocity of the QC
v_max	the maximal velocity of the equipment
v_min	the minimal velocity of the equipment
$\mathbf{v}_p(k)$	the speed vector of the AGV at time instant k
V_c	a finite set of other automata variables of a controlled component
V_uc	a finite set of other automata variables of an uncontrolled component
$\|\mathbf{w}\|_\infty$	the makespan
w_min	the minimal makespan
$\mathbf{x}(k)$	state vector at time instant k
$x_\mathrm{c}^\mathrm{pos}(k)$	the position of a controlled component at time instant k

$x_{\mathrm{c}}^{\mathrm{vel}}(k)$	the velocity of a controlled component at time instant k
X_{c}	a finite set of continuous states of a controlled component
X_{uc}	the set of continuous states of an uncontrolled component
$\mathbf{y}(k)$	output vector at time instant k
$\mathbf{z}(k)$	vector for auxiliary continuous variables at time instant k
$\mathrm{Inv}_{\mathrm{c}}$	the invariant set of a controlled component
$\mathrm{Inv}_{\mathrm{uc}}$	the invariant set of an uncontrolled component
α, β	discrete mode indices
$\delta(k)$	vector for auxiliary integer variables at time instant k
λ	the weighting factor
Φ	a finite set of jobs
Φ_1	a finite set of jobs with dummy job 0
Φ_2	a finite set of jobs with dummy job $N+1$
σ_{ij}^1	sequence variable describing job i and j in Stage 1
σ_{ij}^2	sequence variable describing job i and j in Stage 2
σ_{ij}^3	sequence variable describing job i and j in Stage 3
Ψ_{agv}	a finite set of AGVs
ΔT	the time step size

List of abbreviations

The following abbreviations are used in this thesis:

TEU	Twenty-foot Equivalent Unit
AGV	Automated Guided Vehicle
QC	Quay Crane
ASC	Automated Stacking Crane
MPC	Model Predictive Control
EHMPC	Energy-efficient Hybrid Model Predictive Control
MILP	Mixed Integer Linear Programming

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 100 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.

Xin, J., *Control and Coordination for Automated Container Terminals*, T2015/13, September 2015, TRAIL Thesis Series, the Netherlands

Anand, N., *An Agent Based Modelling Approach for Multi-Stakeholder Analysis of City Logistics Solutions*, T2015/12, September 2015, TRAIL Thesis Series, the Netherlands

Hurk, E. van der, *Passengers, Information, and Disruptions*, T2015/11, June 2015, TRAIL Thesis Series, the Netherlands

Davydenko, I., *Logistics Chains in Freight Transport Modelling*, T2015/10, May 2015, TRAIL Thesis Series, the Netherlands

Schakel, W., *Development, Simulation and Evaluation of In-car Advice on Headway, Speed and Lane*, T2015/9, May 2015, TRAIL Thesis Series, the Netherlands

Dorsser, J.C.M. van, *Very Long Term Development of the Dutch Inland Waterway Transport System: Policy analysis, transport projections, shipping scenarios, and a new perspective on economic growth and future discounting*, T2015/8, May 2015, TRAIL Thesis Series, the Netherlands

Hajiahmadi, M., *Optimal and Robust Switching Control Strategies: Theory, and applications in traffic management*, T2015/7, April 2015, TRAIL Thesis Series, the Netherlands

Wang, Y., *On-line Distributed Prediction and Control for a Large-scale Traffic Network*, T2015/6, March 2015, TRAIL Thesis Series, the Netherlands

Vreeswijk, J.D., *The Dynamics of User Perception, Decision Making and Route Choice*, T2015/5, February 2015, TRAIL Thesis Series, the Netherlands

Lu, R., *The Effects of Information and Communication Technologies on Accessibility*, T2015/4, February 2015, TRAIL Thesis Series, the Netherlands

Ramos, G. de, *Dynamic Route Choice Modelling of the Effects of Travel Information using RP Data*, T2015/3, February 2015, TRAIL Thesis Series, the Netherlands

Sierzchula, W.S., *Development and Early Adoption of Electric Vehicles: Understanding the*

tempest, T2015/2, January 2015, TRAIL Thesis Series, the Netherlands

Vianen, T. van, *Simulation-integrated Design of Dry Bulk Terminals*, T2015/1, January 2015, TRAIL Thesis Series, the Netherlands

Risto, M., *Cooperative In-Vehicle Advice: A study into drivers? ability and willingness to follow tactical driver advice*, T2014/10, December 2014, TRAIL Thesis Series, the Netherlands

Djukic, T., *Dynamic OD Demand Estimation and Prediction for Dynamic Traffic Management*, T2014/9, November 2014, TRAIL Thesis Series, the Netherlands

Chen, C., *Task Complexity and Time Pressure: Impacts on activity-travel choices*, T2014/8, November 2014, TRAIL Thesis Series, the Netherlands

Wang, Y., *Optimal Trajectory Planning and Train Scheduling for Railway Systems*, T2014/7, November 2014, TRAIL Thesis Series, the Netherlands

Wang, M., *Generic Model Predictive Control Framework for Advanced Driver Assistance Systems*, T2014/6, October 2014, TRAIL Thesis Series, the Netherlands

Kecman, P., *Models for Predictive Railway Traffic Management*, T2014/5, October 2014, TRAIL Thesis Series, the Netherlands

Davarynejad, M., *Deploying Evolutionary Metaheuristics for Global Optimization*, T2014/4, June 2014, TRAIL Thesis Series, the Netherlands

Li, J., *Characteristics of Chinese Driver Behavior*, T2014/3, June 2014, TRAIL Thesis Series, the Netherlands

Mouter, N., *Cost-Benefit Analysis in Practice: A study of the way Cost-Benefit Analysis is perceived by key actors in the Dutch appraisal practice for spatial-infrastructure projects*, T2014/2, June 2014, TRAIL Thesis Series, the Netherlands

Ohazulike, A., *Road Pricing mechanism: A game theoretic and multi-level approach*, T2014/1, January 2014, TRAIL Thesis Series, the Netherlands

Cranenburgh, S. van, *Vacation Travel Behaviour in a Very Different Future*, T2013/12, November 2013, TRAIL Thesis Series, the Netherlands

Samsura, D.A.A., *Games and the City: Applying game-theoretical approaches to land and property development analysis*, T2013/11, November 2013, TRAIL Thesis Series, the Netherlands

Huijts, N., *Sustainable Energy Technology Acceptance: A psychological perspective*, T2013/10, September 2013, TRAIL Thesis Series, the Netherlands

Zhang, Mo, *A Freight Transport Model for Integrated Network, Service, and Policy Design*, T2013/9, August 2013, TRAIL Thesis Series, the Netherlands

Wijnen, R., *Decision Support for Collaborative Airport Planning*, T2013/8, April 2013, TRAIL Thesis Series, the Netherlands

Wageningen-Kessels, F.L.M. van, *Multi-Class Continuum Traffic Flow Models: Analysis and simulation methods*, T2013/7, March 2013, TRAIL Thesis Series, the Netherlands

Samenvatting

Automatisering kan de doorstroming van container terminals significant verhogen voor lagere kosten. In geautomatiseerde container terminals worden containers verwerkt door een groot aantal onbemande machines (zoals kadekranen (KKs), automatisch rijdende voertuigen (ARVs) en automatische stapelkranen (ASKs). Deze onbemande machines werken op een interactieve manier met elkaar samen om de containers tussen de kade en het opslaggebied in een terminal te verplaatsen.

Om de prestaties van geautomatiseerde container terminals te vergroten richt dit proefschrift zich op twee aspecten: het verbeteren van de energie efficiëntie en het mogelijk maken van zelfstandiger opererende machines (in het bijzonder vrij-rijdende ARVs)– beide op een operationeel niveau. Aan de ene kant moet de energie-efficiëntie van terminals verbeterd worden, omdat deze onder druk staat door verhoogde energieprijzen en de steeds strakker wordende emissiegrenzen. Aan de andere kant bieden nieuwe type ARVs de mogelijkheid om volledig vrij rond te rijden, waarmee met slimme routeringsstrategiën rijafstanden korter kunnen worden gemaakt dan voorheen. Nieuwe geavanceerde besturingsalgorithmen voor planning en control van die vrij rondrijdende ARVs en gerelateerde machines moeten ontwikkeld worden om die machines optimaal te benutten. Om zowel energie-efficiëntie doelen als betere rijprestaties te realizeren, worden in dit proefschrift discrete gebeurtenis en continue tijd dynamica beschouwd, gebruikmakend van een geintegreerd hybride systeem perspectief.

Voor het analyzeren van de prestaties van voorgestelde methoden richt dit onderzoek richt zich op een compacte, middelgrote, en grote terminal. Voor de compacte en de middelgrote terminal wordt het balanceren van prestaties en energie efficiëntie onderzocht. Voor de grootstalige terminal ligt de focus op het genereren van botsingsvrij paden voor vrij-rijdende ARVs. In het bijzonder worden de volgende aspecten uitgediept:

- De compacte terminal

  Voor de compacte terminal worden de discrete gebeurtenis en de continue tijd dynamica bestuurd door een enkele centrale regelaar. Vanuit het centrale perspectief wordt een hybride modelgebaseerde voorspellende regelstrategie voorgesteld om energie efficiëntie van activiteiten te stimuleren. Het onderliggende optimalisatieprobleem wordt geformuleerd als een gemengd discreet/continue variabelen lineair programmeer probleem, wat efficiënt opgelost kan worden door commerciële oplossers (zoals CPLEX).

- De middelgrote terminal

  Voor de middelgrote terminal neemt de complexiteit aanzienlijk toe door de beschouwing van een groter aantal machines. In vergelijking met de compacte terminal resulteert dit in een groter aantal beslissingsvariabelen, in het bijzonder door de toevoeging van discrete variabelen die gebruikt worden voor het bepalen van zogenaamde opdracht volgordes. Om de complexiteit van de middelgrote terminal handelbaar te maken wordt een hierarchische besturingsarchitectuur voorgesteld. In deze hierarchische architectuur is een automatische bestuurder op een hogere niveau verantwoordelijk voor de discrete gebeurtenis dynamica, terwijl een lagere niveau bestuurder zich bezig houdt met de continue tijd dynamica van elke machine. Zowel zogenaamde open-lus als ook gesloten-lus regel- en coördinatie strategiën worden onderzocht met betrekking tot logistieke prestatie en energie efficiëntie.

- De grootschalige terminal

  Het onderzoeksprobleem dat centraal staat bij de grootschalige terminal richt zich met name op de vrij-rijdende ARVs. Het probleem bestaat uit het bepalen van aanrijdingsvrije paden voor de vrij-rijdende ARVs vanwege veiligheidseisen. Deze paden worden bepaald vanuit een geïntegreerd perspectief, waarbij het plannen van de interacties met andere machines (kranen) centraal staat om een hoge behandelingscapactiteit te kunnen realizeren. Voor dit probleem wordt een sequentiële planningsaanpak voorgesteld. Deze aanpak maakt wederom gebruik van een hiërarchische architectuur waarmee bestuurders op verschillende niveaus gecoördineerd worden. Het gedetailleerde plan van een machine voor een opdracht wordt bepaald aan de hand van een zogenaamde overkoepelende graaf sequentie. Voor een specifieke opdracht omvat die het oplossen van een hybride flow shop planninsprobleem door een hogere niveau bestuurder en een or twee kleinere gemengd discrete/continue variabelen lineair programmeerprobleem door een lagere niveau bestuurder. Met deze aanpak kunnen succesvol aanrijdings-vrije paden voor de vrij-rijdende ARVs worden bepaald.

Samengevat onderzoekt dit proefschrift de operationele besturing van geautomatiseerde container terminals. Het proefschrift toont het potentieel van de nieuw voorgestelde aanpakken voor het verbeteren van de energie-efficiëntie en het faciliteren van de implementatie van vrij-rijdende ARVs.

Summary

It has been demonstrated that automation can significantly increase throughput and reduce cost of container terminals. In automated container terminals, containers are processed by a large number of unmanned machines (e.g., quay cranes (QCs), automated guided vehicles (AGVs) and automated stacking cranes (ASCs)). These unmanned machines are working in an interactive way for transporting containers between the quayside area and the stacking area.

For enhancing the performance of automated container terminals, this PhD thesis focuses on improving energy efficiency and implementing more autonomous equipment (e.g., free-ranging AGVs) at the operational level. On the one hand, due to the increased energy price and environmental stress, energy efficiency needs to be improved. On the other hand, new emerging AGVs allow free-ranging behavior and can shorten the driving distance than using the traditional routing strategy, demanding a novel advanced control algorithm for scheduling and controlling the free-ranging AGVs and the other related machines. For achieving these research goals, both discrete-event dynamics and continuous-time dynamics are considered in this thesis, using a perspective of hybrid systems.

This research focuses on a compact, medium, and large-size terminal. For the compact terminal and the medium-size terminal, the focus is on balancing performance and energy efficiency. For the large-size terminal, the focus is on generating collision-free trajectories for free-ranging AGVs. The following proposals are made in particular:

- The compact terminal

  For the compact terminal, the discrete-event dynamics and the continuous-time dynamics are controlled by a centralized controller as a whole. Using the centralized way, a hybrid model predictive control is proposed for achieving the energy efficiency for real-time operations by controlling each piece of equipment directly. The optimization problem is formulated as a mixed integer linear programming problem that can be efficiently solved by commercial solvers (e.g., CPLEX).

- The medium-size terminal

  When it comes to the medium-size terminal, as the system scale increases, more piece of equipment are considered. Compared to the compact terminal, a large number of decision variables, in particular discrete variables for determining the job sequence that is processed by a particular piece of equipment, have to be considered additionally. Therefore, the complexity for controlling these pieces of equipment grow significantly. To reduce the control complexity of the medium-size, a hierarchical

control architecture is proposed for decoupling the hybrid dynamical system. In this hierarchical architecture, a higher-level controller is responsible for the discrete-event dynamics while the lower-level controller is concerned with the continuous-time dynamics for each piece of equipment. Both the open-loop and the close-loop case have been investigated for achieving the energy efficiency.

- The large-size terminal

  The research problem with respect to free-ranging AGVs is considered in the scope of a large-size terminal, which is the most complex case. It involves collision-free trajectory planning of free-ranging AGVs for safety requirement, integrated with the scheduling of interacting machines for a high handling capacity. For this complex overall problem, a sequential planning approach is proposed. This sequential planning approach uses a hierarchical architecture for coordinating the controllers at different levels and different controllers of AGVs. The detailed schedule is determined job by job following a particular overall graph sequence. For a particular job, the subproblems include a hybrid flow shop scheduling problem at the high-level controller and one or two small-scale mixed integer linear programming problems at the lower-level controller for determining the collision-free trajectory of a free-ranging AGV.

In short, this thesis investigates the operational control of automated container terminals. The thesis shows the potential of the proposed new approaches for improving the energy efficiency and for facilitating the implementation of free-ranging AGVs

Curriculum vitae

Jianbin Xin was born on May 21, 1984 in Zhengzhou, Henan, China. In 2007, he obtained his BSc degree in Electrical Engineering at Xidian University in Xi'an, China. After this, he was recommended to Xi'an Jiaotong University for continuing his master education. He received his MSc degree in Control Science and Engineering in 2010.

From 2010 to 2011, he was working at the Delft Center for Systems and Control at Delft University of Technology, the Netherlands. Afterwards, he moved to the department of Maritime and Transportation Technology at Delft University of Technology and started his PhD project on operational control of automated container terminals. In this PhD project, he is focusing on energy-efficient scheduling and coordination of different types of equipment in automated container terminals. His research interests include model predictive control, hybrids systems and maritime transportation systems. He is a reviewer for IEEE Transactions on Automation and Engineering and Transportation Research Part E: Logistics and Transportation Review.

Publications

1. J. Xin, R.R. Negenborn, G. Lodewijks (2015). Hybrid model predictive control for energy-efficient container handling, *Accepted for International Journal of Control*, DOI:10.1080/00207179.2015.1043350.

2. J. Xin, R.R. Negenborn, G. Lodewijks (2015). Event-driven receding horizon control for energy-efficient container handling, *Control Engineering Practice*, vol.39, pp.45–55, June 2015.

3. J. Xin, R.R. Negenborn, G. Lodewijks (2014). Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control, *Transportation Research Part C: Emerging Technologies*, vol.44, pp.214–230, July 2014.

4. J. Xin, R.R. Negenborn, G. Lodewijks (2014). Trajectory planning for AGVs in automated container terminals using avoidance constraints. In *Proceedings of the 19th IFAC World Congress* (IFAC WC'14), Cape Town, South Africa, pp.9828–9833, August 2014.

5. J. Xin, R.R. Negenborn, G. Lodewijks (2014). Rescheduling of interacting machines in automated container terminals. In *Proceedings of the 19th IFAC World Congress* (IFAC WC'14), Cape Town, South Africa, pp.1698–1704, August 2014.

6. J. Xin, R.R. Negenborn, G. Lodewijks (2013). Hierarchical control of equipment in automated container terminals. *In Proceedings of the 4th International Conference on Computational Logistics* (ICCL 2013), Copenhagen, Denmark, pp.1–17, September 2013.

7. J. Xin, R.R. Negenborn, G. Lodewijks (2013). Hybrid MPC for balancing throughput and energy consumption in an automated container terminal. In *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems* (IEEE ITSC 2013), The Hague, The Netherlands, pp.1238–1244, October 2013.

8. J. Xin, R.R. Negenborn, G. Lodewijks (2013). Hybrid model predictive control for equipment in an automated container terminal. In *Proceedings of the 2013 IEEE International Conference on Networking, Sensing and Control* (ICNSC'13), Evry, France, pp.746–752, April 2013.

9. J. Xin, R.R. Negenborn, G. Lodewijks (2012). Optimal control for coordination of equipment in automated container terminals. In Proceedings of the TRAIL-Beta Congress 2012 — Mobility & Logistics - Science meets Practice, Rotterdam, The Netherlands, October 2012.

10. J. Xin, R.R. Negenborn, F. Corman, G. Lodewijks (2015). Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance, *Submitted to a journal*.