Roboat Formation Control

Sailing in Formation for Waste Transport using Roboat units

M.J. van Pampus

The design of a formation control strategy validated during simulations and real experiments

TU Delft

Roboat Formation Control

Sailing in Formation for Waste Transport using Roboat units

by

M.J. van Pampus

in partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Department Maritime and Transport Technology of Faculty Mechanical, Maritime and Materials Engineering of Delft University of Technology
to be defended publicly on Tuesday July 14, 2020 at 9:30 AM.

Student number:	4224078	
MSc track:	Transport Engineering and Logistics	
Report number:	2020.MME.8441	
Thesis committee:	Prof. dr. R. R. Negenborn,	TU Delft committee Chair
	Dr. V. Reppa,	TU Delft daily supervisor
	Dr. A. Haseltalab	TU Delft daily supervisor
	Ir. V. Garofano	TU Delft daily supervisor
	Dr. L. Ferranti	TU Delft external committee member
	Y. H. Deinema,	AMS Institute committee member
Date:	June 30, 2020	

Preface

Dear reader,

Almost eight years ago I moved to Delft to start a new adventure. I had no idea what a 'student-life' would look like, but I learnt pretty fast I guess. I've had amazing times with all the friends I've had the pleasure to get to know. I mostly did what I wanted to do and felt extremely supported by my parents, sisters, housemates, all the friends I got to know via Virgiel, my 'old' friends from high school, and so on. Together, we did many things that I certainly cannot write down, but about which we've laughed so hard that we will never forget.

Writing down the last couple of words of this thesis feels as if I am moving to another adventure and as has always been the case in these kind of situations, I suffer from a sort of nostalgia. I am ready to move on, but if we could set back the clock for eight years, I would love to do it all again. Before you start reading my thesis, I would like to say some words to you that are on top of my mind.

Somewhat more than a year ago, I started in the RAS lab working on a project with a small vessel, the Delfia 1*. I realized it was a lot of fun and Vittorio and Ali convinced me to continue working on autonomous vessels, this time together with AMS Institute and MIT. It was the right decision and I can't think of a more suitable project for me. Ynse supported me from AMS Institute's side and he made sure every day was full of fun. I won't forget the evening that we were going to drink 'one beer'. By the way I also certainly won't forget the day in the Delftse Hout and the evening that followed with Vittorio, Bernadetto, and Shuo. The PhD defence of Ali including the drinks afterwards was also definitely memorable.

Very special for me was when I got called on a Wednesday evening with the question if I could board a plane to Boston the next morning. Of course I went, got stuck on (the very beautiful) Iceland for a day, had an amazing time at MIT, and got the chance to visit my aunt, uncle, and cousin for the first time. A trip I won't forget, so thanks very much to Rudy, Vasso, Ali, Vittorio, Ynse, Fabio, Meaghan, Rosemeire and all the others that helped me realizing this.

There are so many more people who worked together with me during my graduation and whom I owe a thank you. All colleagues at AMS, but especially Hank, Rens, Stephan, and Ynse. The guys in the RAS lab: Bart, Matt, Wouter, Shuo, Yueming, Bernedetto, and Vittorio. Linying, Ali, Vasso, and Rudy from 'upstairs'. The BEP groups that did projects in the RAS lab, whom I could help sometimes. Then the people from the Senseable City Lab at MIT: Matthew, Wei, Tixiao, Tomas, Pietro, Fabio D., Fabio B., Lorenzo, Martina, Nadinè, Meaghan, Rosemeire and many more. I am sure I forgot to mention some names, but thank you all!

Throughout the complete process of my thesis, most things went pretty smooth. However there were moments I had difficulties seeing where to go and I was lucky with the extensive support I got from Vasso and Ali, and also the support from Vittorio and Ynse. Thanks a lot for that. Near the finish line of my thesis, I lost one of my friends, Mathijs, with whom I had lived together in Delft in a terrible surfing accident. Together with my old housemates in Delft we spent a lot of time together trying to grasp what had happened and to find support in each other. The support I got from all the others around me was overwhelming. While you - Vasso, Ali, Vittorio, and Ynse - were my supervisors, it felt during many moments more like friends and it certainly did feel like that at the time.

Mathijs lived carefree, pointed at the positive sides of life, saw opportunities to have fun everywhere, and has never let a friend down. I envied that very much and will try to adopt his way of life wherever I can.

Life goes on, no matter how painful. I have the privilege to look back on an amazing adventure that lasted for eight years. I gained many friends and lost one. I will take some time off now and start a challenging job at Ahold in October. I am proud of the work I have put in this thesis. But more than that, I am proud of the people I have gathered around me. So a big thank you, to my family, friends, and colleagues for being there always: during the fun, amazing, crazy, and difficult times. I hope you will enjoy reading.

Maurits van Pampus
Delft, June 30, 2020

Executive summary

Automation is seen in all sorts of industries at the moment and the maritime industry is no exception. Many research effort has been put in to make fully autonomous vessels possible. Most applications of autonomous vessels are related to seaworthy container vessels or large barges, however autonomous vessels could also be an essential part of the solution for Amsterdam's household waste problem. Massachusetts Institute of Technology (MIT) and Amsterdam Institute for Advanced Metropolitan Solutions (AMS Institute) have developed a four meters long autonomous boat that can transport the household waste via the canals of Amsterdam: Roboat.

It is estimated that at least 10 Roboat units will be required to transport all household waste of *Amsterdam Centrum* to a collection point. It is desired that this swarm of Roboat units sail in formation whenever possible, mainly for the following three reasons. First, the Roboat units will give less nuisance to other users of the canals. Second, the Roboat units will be easier to observe for other vessels and they can be better monitored, leading to an increased level of safety. Third, a probably sooner authorized solution with one human-operated vessel as leader and autonomous vessels as followers is revealed. The objective of this thesis is to design a strategy that enables multiple autonomous Roboat units to sail in a flexible formation in the canals of Amsterdam for the application of transporting household waste.

In order to reach this objective, first a literature review that discusses the state-of-the-art of maritime formation control and reveals research gaps, is presented. Subsequently, a formation control strategy using 1:4 scale autonomous Roboat units is designed and implemented in Amsterdam's canal network. Finally, the performance of the designed strategy is assessed both in simulations and real experiments.

Literature review

Formation control has been researched on four autonomous vehicle types: ground, aerial, underwater, and surface vehicles. Compared to other sectors, maritime formation control is relatively young and less experiments have been performed in this sector. Important aspects of formation control are: the configuration of formation, the control architecture, and the control method.

The most important characteristic of the configuration of formation is the difference between flexible and fixed configurations. Three main approaches of formation control exist: behavior-based, virtual structures, and leader-follower approaches. Besides, three main control structures exist: centralized, decentralized, and distributed control. Finally, many control methods have been used for formation control.

Striking is that real experiments regarding to formation control have barely been performed with autonomous surface vessels. Predictive control methods have been widely used in simulations, but have not been validated in real experiments regarding to maritime formation control. This is seen as a major research gap.

Formation control strategy design

Based on the widely used guidance, navigation and control (GNC) structure, the fully autonomous control of a single Roboat is accomplished. The dynamics of Roboat are described by the nonlinear differential equation suggested by (Fossen (2011)), and the parameter identification that has been done in (Wang et al. (2018)).

The navigation system is based on the onboard IMU and RTK GPS sensors. The guidance system is split in two systems: a global trajectory planner and a local trajectory planner. While the global trajectory planner determines a global trajectory using user input such as waypoints and a desired speed, the local trajectory planner determines a collision free, reachable reference trajectory using an MILP optimization problem, based on the current position, position of other vessels and / or objects, and the global trajectory. Three control systems have been designed: PI control, model predictive control based on feedback linearization (FL MPC), and nonlinear model predictive control (NMPC).

The control of one Roboat is expanded to a flexible distributed leader-follower formation control strategy by adapting the guidance system of the follower Roboat units. All Roboat units communicate their current and planned future positions in order to establish a train-like or diamond-like configuration. The strategy is

implemented in Roboat using the ROS framework.

Experimental assessment of the designed formation control strategy

Finally, the designed formation control strategies have been experimentally assessed in simulation and real experiments. During all tests, the assigned leader Roboat is given eight waypoints that are ten to twenty meters apart and form an 'eight'-like trajectory that has to be sailed at a speed of 0.2 m/s. In simulation two, and during the real experiments one Roboat units follow the leader. In six simulation tests, all control methods have been tested with full guidance system and with partial guidance system for the leader. In the real experiments this is also done, but only one test with the PI control method is carried out.

Both the FL MPC and NMPC methods give very acceptable results, where the FL MPC method seems to follow the trajectory slightly more accurate, and the NMPC method sails more smoothly and applies less force to the thrusters overall. The FL MPC and NMPC method both outperform the PI method in simulation and during the real experiments. The average value of normalized KPIs shows that the NMPC method overall performs circa 20% better than the FL MPC method in practice, while the overall performance for these two methods in simulation is similar.

The guidance system works well in general, but during object avoidance application, the calculation time became 2.5 s, where 0.5 s is the critical value to use in practice. Implementation of the algorithm on the Roboat units, written in C++ code instead of being implemented on a central computer in MATLAB should reduce the calculation time drastically.

In summary, the results of this thesis indicate that the designed formation control strategies work sufficiently well and can be implemented on full scale Roboat units with some modifications. To the best knowledge of the authors, the real experiments with predictive maritime formation control strategies are a first. This is the main contribution of this thesis. Possibilities for future research exist in multiple directions. The global operation of waste collection using Roboat units can be worked out more, the control methods can be improved, and the designed strategies can be coded more efficiently.

Nomenclature

Greek symbols

Δ	Change in subsequent symbol
ϵ	Surge speed [m/s]
ϵ_d	Reference surge speed [m/s]
ζ	Input vector of linearized system
η	Vector of position
$\hat{\eta}$	Vector of measured position
η_d	Reference trajectory (matrix containing desired positions per time step)
$\tilde{\eta}_d$	Global reference trajectory
ν	Vector of velocity
$\hat{\nu}$	Vector of measured velocity
$\overline{\nu}$	Vector of average velocity
τ	Vector of internal forces and moments
τ_d	Vector of reference internal forces and moments
τ_d^{unsat}	Vector of unbounded reference internal forces and moments
τ_{min}	Vector of minimum internal forces and moments
τ_{max}	Vector of maximum internal forces and moments
ψ	Heading, component of position vector [rad]
$\hat{\psi}$	Measured heading [rad]
ψ_d	Reference heading [rad]
$\overline{\psi}$	Average heading
Ω	Sensor data

Latin symbols

A	State matrix of linearized system
A_d	Discretized state matrix of linearized system
a	Distance between the transverse thrusters
$\mathbf{B}$	Thrust allocation matrix
B	Input matrix of linearized system
B_d	Discretized input matrix of linearized system
b	Distance between the longitudinal thrusters
$\mathbf{C}$	Matrix of Coriolis and centripetal terms
$\mathbf{D}$	Drag matrix
$\mathbf{d}$	Vector of external forces and moments
d_{safe}	Minimum required safety distance between center of gravity of own vessel and other vessels or objects
$\mathbf{e}$	Vector of position error relative to reference position
i	Numbering of prediction step
J	Objective function
j	Numbering of vessels and objects; auxiliary numbering
$\mathbf{K}_d$	Derivative gain of PID controller
$\mathbf{K}_i$	Integral gain of PID controller
$\mathbf{K}_p$	Proportional gain of PID controller
k	Discrete time step
$\mathbf{M}$	Matrix of added mass and inertia
m	Number of vessels and objects
m_{11}	Component of the added mass and inertia matrix
N_p	Number of prediction steps

N_r	Component of the drag matrix
N_{TA}	Number of thrust allocation prediction steps
n_{1j}	Desired discrete headway between vessel 1 and j [-]
n_a	Number of actuators
$\mathbf{P}_{NMPC}$	Penalty matrix for NMPC
$\mathbf{P}_{TA}$	Penalty matrix for thrust allocation
$\mathbf{p}_j$	Vector of position without heading of vessel or object j
$\hat{\mathbf{p}}_j$	Vector of measured position without heading of vessel or object j
$\tilde{\mathbf{p}}_j$	Global reference position without heading of vessel or object j
$\mathbf{Q}_{FL}$	Penalty matrix for FL MPC
$\mathbf{Q}_{NMPC}$	Penalty matrix for NMPC
$\mathbf{Q}_{TA}$	Penalty matrix for thrust allocation
$\mathbf{R}$	Rotation matrix
r	Yaw rate [rad/s]
T	Total time of test
t	Continuous time [s]
t_{1j}	Desired headway between vessel 1 and j [s]
t_s	Sampling time [s]
$\mathbf{u}$	Vector of control input
$\mathbf{u}_{\min}$	Vector of minimum input of actuators
$\mathbf{u}_{\max}$	Vector of maximum input of actuators
u_1	Control input of thruster 1
v	Sway velocity [m/s]
$v_{\min}$	Minimum speed [m/s]
$v_{\max}$	Maximum speed [m/s]
Wp	Matrix containing waypoints
X_u	Component of the drag matrix
$\mathbf{x}$	State vector containing the position and velocity
x	Component of position vector; in NED frame: North, in body-fixed frame: Surge [m]
x_d	Reference x component [m]
Y_v	Component of the drag matrix
y	Component of position vector; in NED frame: Easth, in body-fixed frame: Sway [m]
y_{1j}	Desired lateral distance between vessel 1 and j [m]
y_d	Reference y component [m]

Superscripts

b	Body-fixed frame
n	NED frame
T	A transpose is taking place

List of Figures

List of Tables

Contents

1
Introduction

In today's world the need for automation is present in almost all industries and processes. Cars are being welded together by robots, algorithms trade securities in milliseconds on the stock markets and the scheduling of Uber's taxi rides is based on machine learning, to name a few. Benefits of automation include costs savings, reduction of errors, and an improvement of safety. These benefits are recognized in the maritime industry as well and many research effort has been made to reduce the amount of necessary personnel on vessels and to make fully autonomous trips possible in the near future.

While the technology for autonomous vessels could be helpful for large container ships crossing the ocean, it could also be an essential part of the solution for an urgent problem Amsterdam has to deal with. In the busy city center of Amsterdam, household waste is dumped onto the streets and collected by waste trucks that cause damage to the historic and iconic bridges and quay walls, among other things. A solution to transport the household waste via the canals of Amsterdam using a small-sized, autonomous vessel has been proposed by Massachusetts Institute of Technology (MIT) and Amsterdam Institute for Advanced Metropolitan Solutions (AMS Institute). This thesis contributes to this solution by exploring the possibilities of formation control of a fleet of these so-called Roboat units.

In this chapter first the background of the problem will be discussed. Thereafter the problem statement is explicitly addressed, which includes the forming of the research questions and research approach. The chapter concludes with the outline of the remaining part of the report.

1.1. Background

This thesis aims to contribute to the solution of the Amsterdam waste problem by improving autonomous shipping technology applicable to the Amsterdam city center area. In order to have sufficient background knowledge to understand the complete problem and determine the areas that need more research, this section provides context on different parts of the problem and solution. First of all the waste collection in Amsterdam will be addressed. Secondly, background information on autonomous shipping applied to the problem of waste collection in Amsterdam is provided. Then, the current state and performed research on Roboat is analyzed. This section concludes with the area that is found to be in need of more research: formation control.

1.1.1. Waste collection in Amsterdam's city center

In the current situation all 86,862 residents of city district *Amsterdam Centrum* [1] may place their household waste on the street two times a week (Gemeente Amsterdam (2019)). Residents of Amsterdam produce on average 301.4 kilograms of household waste per year[2] (CBS (2019)). This means that every year about 26,200 MT of waste is being collected in *Amsterdam Centrum*. Per week this is on average 503 MT, similar to 100,000 garbage bags of 5 kg each per week. Since the total land area of *Amsterdam Centrum* is only 6.28 km^2 (OIS Amsterdam (2018)) - circa 950 football fields - the average amount of garbage bags per week is about 105 per football field. Realize that on that football field, a lot of space is filled by houses already. The garbage bags

[1] determined at January 1, 2019 (OIS Amsterdam (2019))
[2] bulky household waste, coarse garden waste, wood waste (A and B wood) and clean debris excluded

Figure 1.1: Pile of garbage bags in *Amsterdam Centrum* ©2019 AT5

pollute the city and destroy its image. Furthermore the garbage bags attract seagulls, rats and other vermin. An example of Amsterdam's streets polluted by garbage bags is shown in Figure 1.1.

On request the manager of industrial waste collection of the municipality of Amsterdam explained that on average 9 waste trucks, weighing 20 MT when empty, to 28 MT when full, drive for eight hours per day, six days per week to collect the waste in *Amsterdam Centrum*. First of all these trucks hinder traffic and cause nuisance to residents. Furthermore the trucks damage the historic, fragile bridges and quay walls, designed to carry the weight of horses and carriages. The municipality has budgeted 450 million euro to renovate 27 bridges and 800 meters of quay walls before 2024, however it is estimated that 750 bridges and 135 kilometres of quay walls do not meet the legal requirements and need to be renovated (AT5 (2019)). These are extraordinary costs that are partly caused by the waste trucks.

The total costs of the waste collection for *Amsterdam Centrum* are not clear. However based on different sources, it is roughly estimated that the total costs of waste collection for *Amsterdam Centrum* are about 3.0 million euro per year. The estimation of the total costs of the waste collection is substantiated in Appendix C. It is emphasized that the estimation provides only an indication.

1.1.2. Autonomous shipping

Right now is an exciting period of time for the autonomous shipping industry. The tipping point after which autonomous vessels will be deployed in commercial tours is approaching. In December 2018 Rolls-Royce and Finferries have successfully demonstrated the world's first fully autonomous ferry in the archipelago south of the city of Turku, Finland (Rolls-Royce Holdings plc (2018)). In the experiment with the car ferry *Falco* no human intervention was necessary.

While the test with *Falco* was an experiment, some of the commercial vessels may do their daily work crewless in 2022 the earliest. *Yara Birkeland*, a 260-foot, fully electric and autonomous container ship currently being developed by Kongsberg, will follow short and specific paths through Norway, carrying chemicals and fertilizer from a production plant to nearby towns (Furness (2019); Muzaffar (2019)). The full autonomous trips are planned to happen in 2022 (Kongsberg Maritime (2020)). Other companies currently working on autonomous vessels are, among others, Wilhelmsen, iXblue, Aker Arctic, Fugro, L3 ASV, and Samskip (MI News Network (2018)).

Different definitions for autonomous vessels exist in literature. When spoken of autonomous vessels in this research, a level of autonomy that is at least that of the definition given by (MUNIN (2016)) must be present. That definition is the following: [An autonomous vessel is a] "vessel primarily guided by automated on-board decision systems but controlled by a remote operator in a shore side control station." This implies that an autonomous vessel can be controlled without any operators on board. Therefore unmanned vessels and autonomous vessels will have the same meaning in this research, unless explicitly stated otherwise.

Advantages of autonomous shipping It seems inevitable that autonomous vessels will appear and that is not without reason. Autonomous vessels are beneficial over human-operated vessels for a number of reasons from which improved safety is one of the main advantages. It is estimated that human errors are responsible for between 75% and 96% of marine casualties (Allianz Global Corporate & Specialty (2012); Dean et al. (2018)). However, (Dean et al. (2018)) rightly notes that little has been said or published about the number of accidents avoided by human intervention. Besides, despite autonomous vessels effectively removing human error, they will not bring the risk to zero (Furness (2019)). Nevertheless they do allow for operations that do not put human lives at risk (Muzaffar (2019)).

Another main advantage of autonomous vessels is the economic benefit, as operational costs will reduce without crew members on board. Besides, vessels could be redesigned to be more efficient, since ship builders can eliminate accommodation structures such as the deckhouse and living quarters, as well as minimize energy expensive functions such as heating and cooking facilities (Furness (2019)).

Transformation to autonomous shipping In (Ovcina Mandra (2018)) the CEO of Maersk Line, Mr. Soren Skou, stated that it is not expected that ultra large container vessels (ULCV) will be allowed to sail without humans on board in the near to mid-term future, for the potential economic savings are marginal. For smaller container vessels or other cargo vessels the total transport cost savings can be 10 to 22 percent by switching to unmanned operation, according to (Ovcina Mandra (2018)). This group, that mostly sails on the inland waterways, is therefore the most likely group that will be the first to make commercial use of autonomous vessels.

While the use of an autonomous vessel alone already seems beneficial over using a human-operated vessel, autonomous vessels also introduce new possibilities in terms of smart cooperation between vessels and infrastructure. Communication between two vessels that encounter each other could be intensified when the communication is autonomous, allowing for a safer operation due to the knowledge of the intentions of the other vessel. Speed of the vessels can be optimized to be exactly on time for the opening of ports and locks if the vessels can communicate with infrastructure (Chen et al. (2019c)). Last but not least vessels can form a formation of vessels that sail in the same direction, taking up less space of the water and minimize for example the amount of time that bridges and locks need to be used.

1.1.3. Using autonomous vessels for waste collection

MIT and AMS Institute are working on a solution for the Amsterdam waste problem that consists of transporting household waste via the canal network using small autonomous vessels. The main idea is that autonomous vessels transport the household waste from collection points near or on the canals in *Amsterdam Centrum* to the incinerator of waste processing company *AEB* that is located near the water. The impact of the solution to transport waste via Amsterdam's canal network using autonomous vessels will be discussed in this subsection.

In the last years, three different scenarios to collect the waste on the collection points using an autonomous vessel have been devised:

1. Residents can dump their waste in floating dumpsters at a specified time and place, twice a week (similar to the current situation). The floating dumpsters will be replaced by an autonomous vessel when full and will move to another location on an in advance specified time.

2. Residents can dump their waste in semi-permanent floating dumpsters every day. The floating dumpsters will be replaced by an autonomous vessel when full.

3. Residents can dump their waste in containers built in the quay wall. When the container is full an autonomous vessel will sail to the container. Subsequently the waste can be dumped into the autonomous vessel via an opening in the quay wall just above the water.

The master thesis of (Daub (2019)) contains research using scenario 1. In this research, it was assumed that the collected waste will be transferred to a fictional storage unit near *Amsterdam Centrum.* From the fictional storage unit the waste has to travel another 14 km to the incinerator from the waste processing company *AEB* via the waterways *IJ* and *Noordzeekanaal.* The areas of interest are shown in the map in Figure 1.2. In the current situation no solution to transport the waste from a dumpster or Roboat (more information is given in Section 1.1.4) to the storage unit has been worked out. One idea is to use a crane to lift a separate container that contains the waste and empty it on a conveyor belt that transports the waste to the incinerator.

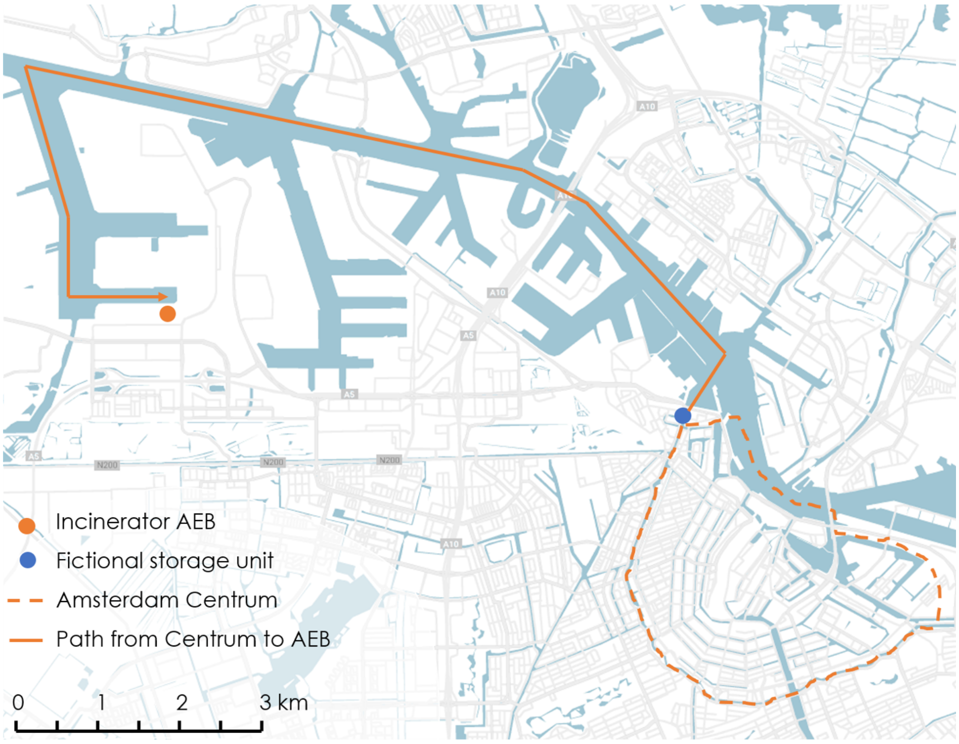


Figure 1.2: *Amsterdam Centrum*, the fictional storage unit and the incinerator

A second idea is to design a mechanism in the dumpster or Roboat that is able to lift and tilt the container from the vessel to empty it on a conveyor belt.

In the current situation also no solution to transport the waste from the storage unit to the incinerator has been worked out. A first idea is to leave out the storage unit and to sail directly to the incinerator of *AEB*. Another idea is to sail the part between the storage unit and the incinerator of *AEB* once or twice a day with a larger barge.

Impact of transporting waste via canals using autonomous vessels Transporting waste via the canals of Amsterdam will differ compared to the current situation. The expected results of transporting waste via the canals of Amsterdam are shown in Table 1.1. The expected results of transporting waste with autonomous vessels instead of human-operated vessels are listed in Table 1.2.

Table 1.1: Expected results of transporting waste via canals compared to road

	Expected advantages	Expected disadvantages
1	No garbage bags polluting the streets of Amsterdam	Since waste is placed near the water, the canals can be polluted
2	No waste trucks blocking the streets of Amsterdam and creating noise	Due to an increase in traffic via the canal network, the canals could become congested
3	No waste trucks damaging the bridges and quay walls	
4	Residents can dump their waste every day of the week [a]	

[a] Depending on the chosen solution.

Cost estimation The three scenarios are still under investigation and at the moment no cost estimation exists. However in the master thesis of (Daub (2019)) including references, some information is provided about

Table 1.2: Expected results of transporting waste using autonomous vessels compared to human-operated vessels

	Expected advantages	**Expected disadvantages**
1	Labour costs will be lower since no personnel is necessary	The purchase and / or development costs will be higher
2	Waste could be collected during night time - if the process of collecting makes noise of an acceptable level - without any extra costs compared to collection during day time	The darkness and weather can impact the sensor operation
3	If multiple vessels are being deployed to collect the waste, they can work together	Garbage bags placed next to a container or floating platform cannot be collected
4	The amount of human errors will be reduced	

(a) 1:4 scale Roboat on the quay in Amsterdam near AMS Institute (b) 1:2 scale Roboat just before going into the water ©2019 AMS Institute (c) Render of full scale Roboat transporting a dumpster ©2018 Pietro Leoni

Figure 1.3: Different versions of Roboat

the number of necessary Roboat units and floating dumpsters on which, together with some assumptions, a very rough cost estimation has been made. In Appendix C the complete substantiation of this estimation can be found. In summary, it is estimated that 10 Roboat units and 96 dumpsters will be sufficient to collect all household waste of *Amsterdam Centrum* under aforementioned scenario 1: residents can dump their waste twice a week in floating dumpsters. The yearly total costs are estimated from around 500,000 euro without extra operators to 900,000 euro with two operators monitoring the Roboat units all day long (24 hours per day).

It is emphasized that the estimation of the costs of waste collection with a Roboat is based on assumptions that could not be verified. Besides, only a cost estimation for scenario 1 is given, since no extensive research to scenarios 2 and 3 has been performed yet. Moreover, currently it is assumed that the solution to transport the waste from the dumpster or Roboat to the storage unit and from there to the incinerator does not cost extra money or personnel. This is unlikely, but it is not possible to estimate the extra costs of this, since no solution is worked out at the moment. However the cost estimation gives the indication that waste transport using Roboat units could be financially viable since the total costs of the current collection of waste has been estimated at 3.0 million euro per year in Section 1.1.1.

1.1.4. Roboat

Roboat is an autonomous vessel recently developed by the Massachusetts Institute of Technology (MIT) in collaboration with the Amsterdam Institute for Advanced Metropolitan Solutions (AMS Institute). The project investigates different possibilities with a fleet of autonomous floating vessels for the city of Amsterdam. AMS Insitute possesses one full scale Roboat, with dimensions of 4 by 2 meters. Besides the full scale, also 1:2 and 1:4 scale Roboat units exist with dimensions of respectively 2 by 1 and 1 by 0.5 meters. In Figure 1.3 pictures of the different versions of Roboat are shown.

In the last years a lot of research activity has been carried out using the 1:4 scale Roboat units. A tracking control strategy using real-time nonlinear model predictive control is developed in (Wang et al. (2018)). In (Mateos et al. (2019)) an autonomous latching system is discussed, while in (Park et al. (2019)) a method for a reconfigurable multi-vessel platform is reviewed. Furthermore the master thesis described in (Gheneti

(2019)) contributed by designing LiDAR-based perception algorithms and trajectory optimization algorithms for multi-robot systems. The latter is also described in (Gheneti et al. (2019)). The algorithms that have been designed for multi-robot systems, assume one 'smart' Roboat equipped with sensors and computational power, and multiple 'dumb' Roboat units that have limited sensors and computational power. The idea is that the 'smart' Roboat determines the actions of the 'dumb' Roboat units to form reconfigurable platforms. In Table 1.3 the possible research opportunities regarding autonomous vessels are shown, together with the way they have been addressed in the Roboat project.

Table 1.3: Possible research opportunities regarding autonomous vessels addressed by the Roboat project

Research area	Research opportunity	Published	Developed / In development	Not yet addressed
Guidance	Path planning	Wang et al. (2018)	X	-
	Trajectory planning	-	X	-
	Latching to other Roboat units	Mateos et al. (2019)	X	-
Sensors	Object detection	-	X	-
	SLAM[a]	-	X	-
	Sensor fusion[b]	-	-	X
Control	Path following	-	X	-
	Trajectory tracking	Wang et al. (2018)	X	-
	Obstacle avoidance	-	X	-
	Adaptive control (weight distribution)[c]	-	X	-
Muti-vessel Control	Adaptive control (multi-vessel platform)[d]	Gheneti et al. (2019); Gheneti (2019); Park et al. (2019)	X	-
	Formation control	-	-	X

[a] Simultaneous localization and mapping
[b] For example the information of SLAM using a LiDAR and the information of a GPS unit
[c] A control algorithm that is able to adapt depending on the weight distribution
[d] A control algorithm that is able to adapt depending on the configuration of multiple Roboat units attached to each other

As seen in Table 1.3, formation control, or collaboration between multiple 'smart' Roboat units has not been studied yet. This would be very helpful in the case that Roboat units will work in the same small area to deliver and pickup dumpsters. In Figure 1.4 the proposed locations of the dumpsters are shown. Since a Roboat can only transport three dumpsters and the combination of Roboat and dumpsters cannot make a 180 degrees turn in the canals, most of the deployed Roboat units will sail the same route. If Roboat units work together, they could sail along some routes in formation, ensuring that other users of the canals experience less nuisance.

1.1.5. Formation control

For the vessels that sail on the inland waterways, an interesting opportunity is the cooperation of multiple autonomous vessels. Advantages of multi-vessel cooperation include robustness, reliability and efficiency. For instance, with cooperation, vessels can coordinate their voyage plans to avoid congestions at ports and locks, enhancing the robustness of waterway networks (Chen et al. (2018)). Besides, communicating vessels can, amongst others, share their intentions, and information about objects beyond the reach of sensors of one of the vessels, enhancing the reliability of the vessels' systems (Chen et al. (2018)). Moreover, with cooperation, a group of vessels can carry out tasks such as search and rescue or ocean sampling more efficiently (Chen et al. (2018)). In the next paragraphs an introduction of vessels that cooperate and sail in a formation will be given.

Vessels can be kept in formation via manual control, remotely manual control, or autonomous control. In this research the latter will be handled. A form of autonomous formation control is the Vessel Train (Novimar

Figure 1.4: Proposed locations of the dumpsters divided in six zones ©2019 Snoweria Zhang

(2017)). In this concept a human-operated vessel is the leader of the Vessel Train. Several autonomous vessels follow the human-operated leader, and hence require less personnel. Other situations that would benefit from autonomous formation control are transportation in urban areas, large object transport, surveillance, search and rescue, etc. (Chen et al. (2019b)).

For the case of using autonomous vessels to solve the waste problem in Amsterdam, formation control is mainly desired for three reasons. First, the autonomous vessels will give less nuisance to the other users of the canals, since they sail close to each other in formation. Second, the small autonomous vessels will be easier to observe for other vessels and they can be better monitored, leading to an increased level of safety. Third, a probably sooner authorized solution with one human-operated vessel as leader and autonomous vessels as followers is revealed.

Formation control is also being developed in other industries, such as the aerial and road industry. The expected benefits of formation control of trucks, often called platooning, include increased fuel and traffic efficiency, safety, and driver comfort (Bergenhem et al. (2012)). The reduction of overall fuel consumption of the trucks can be up to 20%, according to (Liang et al. (2016)). Fuel savings are also in the aerial industry drivers for formation control (Zhang and Liu (2019)). In addition formation control of unmanned aerial vehicles is advantageous in the military domain (Yu et al. (2013)). A more detailed literature review of formation control can be found in Chapter 2.

1.2. Problem statement

As previously described, Amsterdam has to deal with a waste problem. A possible solution would be to transport household waste via Amsterdam's canals using autonomous Roboat units. This research focuses on the autonomous shipping part of that solution, and contributes to formation control. In this section, the research objective, research questions, and research approach are discussed.

Definition of formation control In this research the definition of formation control that is proposed by (Schoerling et al. (2010)) will be used: *'In formation control, a group of cooperative vehicles must keep some user-defined distances to each other, while performing a maneuver'*. Similar to the use of definitions in (Chen et al. (2018)), a formation will be called *flexible* if it is allowed to temporarily change the user-defined distances between vehicles. This could be useful to avoid objects for instance.

1.2.1. Research objective

The objective of this research is to design a strategy that enables multiple autonomous Roboat units to sail in a flexible formation in the canals of Amsterdam for the application of transporting household waste. This research will not investigate the problem of collecting the dumpsters or waste by the Roboat units. In other words, only a formation of empty Roboat units without dumpsters will be considered. The proposed strategy will be validated in computer simulations and in real experiments using 1:4 scale Roboat units. This objective has been formed into the following main research question:

How to design and implement a flexible formation control strategy in Amsterdam's canal network using autonomous Roboat units?

1.2.2. Research questions

To be able to answer the main question, research questions have been formed. During the remaining part of this research, the research questions will be answered by literature studies, computer simulations and real experiments. The research questions have been formed in such a way that the main question will be answered after all research questions have been answered. The complete list of research questions is the following:

1. What is the current state of research on maritime formation control?
   - (a) In what applications other than maritime applications is formation control being used?
   - (b) What kind of research has been done on formation control?
   - (c) What area of maritime formation control can be improved?
2. How to design a flexible formation control strategy?
   - (a) How to control a single Roboat fully autonomously?
   - (b) What control structure for multiple Roboat units should be used?
     - i. What configuration of formation to use?
     - ii. What flexible control strategy to use?
     - iii. How to communicate between multiple Roboat units?
3. How to implement the formation control strategy using Roboat units?
   - (a) What are the properties important for implementation and what are the required specifications?
   - (b) What communication and command structure to implement?
   - (c) How can the characteristics of Amsterdam's canal network be considered in computer simulations and real experiments?
4. What is the performance of the designed formation control strategy and its implementation in Amsterdam's canal network?
   - (a) Which indicators are frequently used to measure the performance of a formation control strategy?
   - (b) What is the performance of comparable systems in literature?
   - (c) How can the chosen performance indicators be tested?
   - (d) What is the performance of the formation control strategy, based on the chosen performance indicators?

In each chapter the research question that will be answered, will be mentioned in the introduction part.

1.2.3. Research approach

To address the aforementioned research questions, the following approaches are used. First of all, the state-of-the-art of maritime formation control is revealed by categorizing existing research and identifying research gaps.

Secondly, three control strategies for one autonomous vessel are designed. The design of the baseline control strategy is based on literature. The other two strategies are designed to be compatible with a formation control strategy and supported by literature.

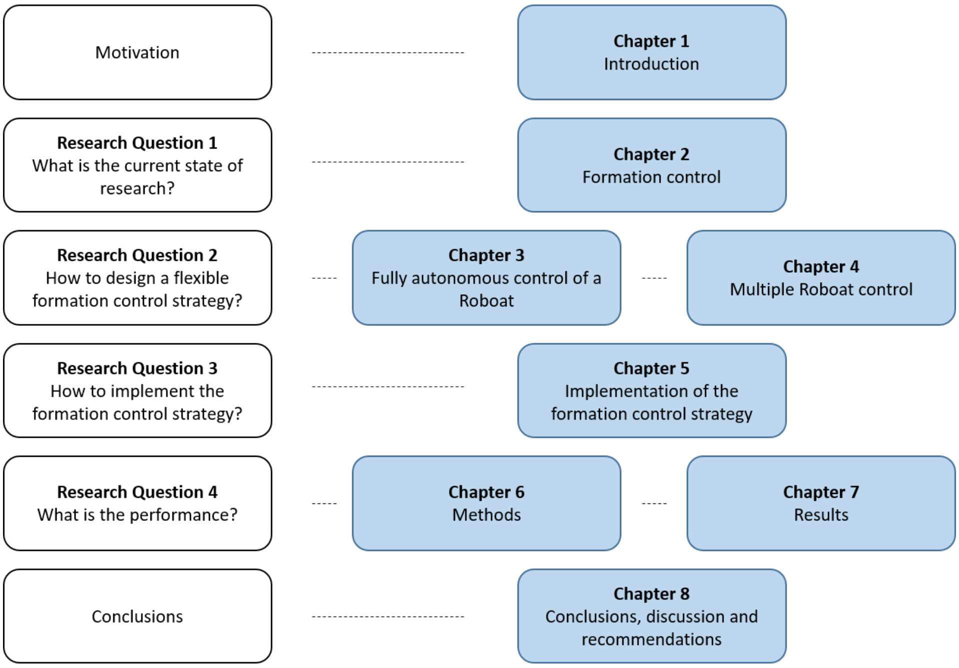


Figure 1.5: Outline of this thesis

After the validation, the control strategies for one vessel are expanded to a formation control strategy. The aim of this design is to fill the research gaps that are identified during the literature review (see Chapter 2).

Finally, the designed strategies are assessed during simulations and real experiments using 1:4 scale Roboat units.

1.3. Thesis outline

In the following paragraphs, the organization of the remaining part of this thesis is described. A schematic overview of this is also shown in Figure 1.5. First, Chapter 2 will discuss the state of the art of maritime formation control, and will give an answer to research question 1.

Then, focus will shift to the design of the formation control system. In Chapter 3 a system to control one Roboat is designed, answering research question 2a. Thereafter, in Chapter 4, the control system is expanded to control multiple Roboat units while performing formation control. In this chapter, research question 2b is answered. Subsequently Chapter 5 discusses the implementation of the designed control system using Roboat units. This will provide an answer to research question 3.

Afterwards, the designed control system is experimentally assessed during simulations and real experiments. In Chapter 6, the methods used will be analyzed and an answer to research question 4a, 4b, and 4c will be given. Subsequently the results of both simulations and real experiments will be shown in Chapter 7, answering research question 4d. Finally, conclusions will be drawn and recommendations for future research will be given in Chapter 8.

2
Formation control

The first step towards a formation control design, is to consult conducted research works in this area. In this chapter the relevant contributions that have been made to the field of formation control will be discussed. The research question that will be answered in this chapter is *'What is the current state of research on maritime formation control?'*. This question is divided into the following questions:

a) In what applications other than maritime applications is formation control being used?
b) What kind of research has been done on formation control?
c) What area of maritime formation control can be improved more?

Existing literature on formation control will be methodologically analyzed in order to understand the factors involved in formation control. Also a look into formation control in aerial and ground applications will be given, since the development of formation control in those sectors is ahead of the maritime sector. Finally, the research gaps in maritime formation control are presented. This research will try to fill one of these research gaps in the remaining parts.

2.1. Relevant research on formation control

In order to create a better understanding of formation control and the various options that can be chosen to control a formation, the content of former research works will be analyzed in this section. The main focus will be on maritime formation control, but also research works on formation control using autonomous ground vehicles and autonomous aerial vehicles will be included in the analysis.

Relevant research works will be compared on several aspects related to formation control. Each aspect will be discussed in a separate section. The aspects covered are the following:

- **Configuration of formation**: the relative positions of the vehicles;
- **Control architecture**: the framework for exchanging information and making decisions for the complete formation as well as the single vehicles;
- **Control method**: the way in which a reference value - for instance a reference trajectory - is converted into actions enabled by the actuators based on sensor measurements.

2.1.1. Configuration of formation

A group of trucks, vessels, or quadrotors can all be operating in formation. The purpose for which the formation is set up could differ, and is important for the design of the formation control strategy. The main goal of a formation of trucks is to reduce air resistance. In that case, the formation of the trucks resembles a so-called road train. A task of the quadrotors is to lift something from the ground. In that case, the formation needs to be fixed. In this section different configurations of formation consisting of different types of vehicles will be discussed and linked to formations with vessels. The most important difference between configurations of formation is that the configuration can be flexible or fixed. All research results that are discussed in this section, are summarized at the end of this section in Table 2.1.

Figure 2.1: Situation that requires a fixed configuration ©2017 Hebron Project

2.1.1.1. Fixed configuration

In Figure 2.1 an example is shown where a fixed configuration is required for the formation. In this example, the towing vessels need to be on specific positions to transfer the right amount of forces to the enormous platform of about 220 meters high (ExxonMobil (2015)). In the case of a fixed configuration, the formation needs to act as one to make sure that the configuration remains the same in all cases. An example of the design of a formation control system for fixed configurations can be found in (Breivik et al. (2008)), where simulations have been performed with three ASVs. The goal of their triangle-shaped, fixed-configuration is to explore and exploit hydrocarbon in Norway in the future. More recently, in (Chen et al. (2019b)) a formation control system for fixed configurations is designed and validated through simulations of three ASVs. Although the control architecture and method are completely different compared to the methods used by (Breivik et al. (2008)) - which will be discussed later on -, the configuration of formation was the same. The three ASVs were used to transport a floating object, while being fixed to the object with towlines.

2.1.1.2. Flexible configuration

The counterpart of the fixed configuration, is the flexible configuration. In Figure 2.2 an example of a flexible configuration is shown. In the figure a platoon of a truck and three cars can be seen. The truck and cars can leave the platoon if they want and likewise other trucks and cars can join the platoon. In a flexible configuration the distance constraints are softer than in a fixed configuration.

An example of a flexible configuration can be found in (Alonso-Mora et al. (2018)), which describes the flexibility of the formation of four autonomous quadrotors. In the experiment, a human came into the flying zone of the quadrotors that flew in a square formation. The quadrotors changed the configuration of the formation to avoid the human and after the human left, the quadrotors returned to their original positions and configurations. Other research results on flexible configurations can be found in (Ogren (2004); Raghuwaiya et al. (2016); Zhu et al. (2019)).

In (Ogren (2004)), vehicles travel along their own paths, but when they find themselves moving close to another vehicle, the vehicle switches to follower mode for as long as the other vehicle is beneficial to it. The research of (Raghuwaiya et al. (2016)) proposes a method to split a formation in order to avoid objects and afterwards perform a rejoin maneuver. The quadrotors used in (Zhu et al. (2019)) are controlled in a way that they can split the formation to avoid obstacles, or in case of failure of one of the vehicles in the formation. Afterwards, the vehicles are able to merge again. In general a flexible configuration offers more options to deal with the (dynamic) surroundings compared to a fixed configuration.

Figure 2.2: Flexible configuration: cars can join and leave the platoon ©2012 Volvo Car Corporation

Vessel Train Formation In the maritime sector a formation of vessels on inland waterways will likely result in train-liked formation due to the limited boundaries of waterways (Chen et al. (2018)). This is also conform the Dutch legislation for inland waterways, which states that vessels should remain on the starboard side of the water in most waterways and that sailing next to each other is only allowed if it causes no hindrance or danger to other vessels (Rijkswaterstaat (1983)). This configuration, with vessels behind each other, separated by at least the safety distance, is called the Vessel Train Formation (VTF). The VTF is the subject of research in (Chen et al. (2019c)) and earlier work. The VTF is flexible, since vessels can join and leave formations all the time. Vessels are also allowed to overtake other vessels in the formation. The VTF is also being researched by (Novimar (2017)), but in their research the leader is a human-operated vessel. In this case, vessels cannot overtake other vessels and are not allowed to leave the formation.

Strict conditions Another special case of the flexible configuration, is the flexible configuration with strict conditions. In (Alonso-Mora et al. (2017)) an experiment with three mobile manipulators (ground vehicles) that collaboratively carry an object is successfully conducted. In that experiment the original formation of the mobile manipulators was an equally distributed tripod. However the configuration changed slightly when the team had to go through a narrow part, while still being able to carry the object.

Table 2.1: Configuration types in formation control of autonomous vehicles

Sector	Type of vehicle	References	Configuration of formation	Validated by
Ground vehicles	Unicycle robots	Ogren (2004)	Flexible	Simulation
	Mobile robots	Raghuwaiya et al. (2016)	Flexible	Simulation
Aerial vehicles	Quadrotors	Alonso-Mora et al. (2017)	Flexible (strict conditions)	Simulation & Experiments
	Quadrotors	Zhu et al. (2019)	Flexible	Simulation & Experiments
Surface vehicles	Vessels	Breivik et al. (2008)	Fixed	Simulation
	Vessels	Chen et al. (2019b)	Fixed	Simulation
	Vessels	Chen et al. (2019c)	Flexible (VTF)	Simulation

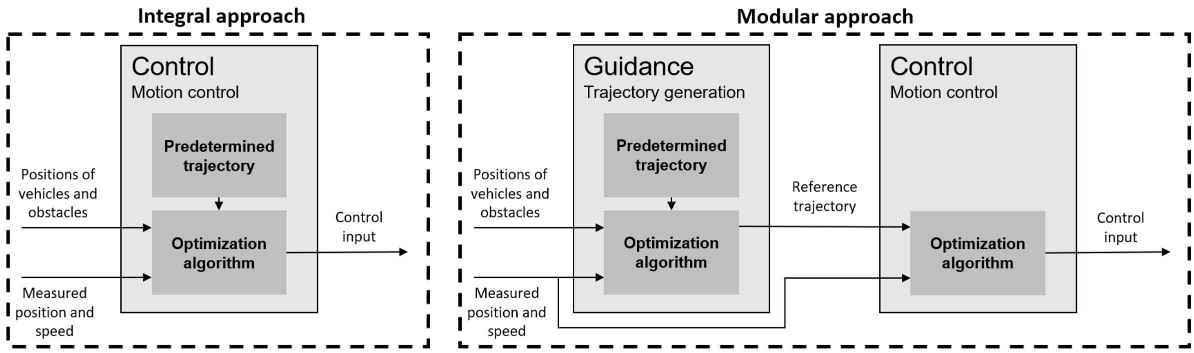


Figure 2.3: Schematic overview of the integral and modular approach for the control of a single vessel

2.1.2. Control architecture

When vessels sail in a formation, the formation moves as a whole whenever possible. In a flexible configuration, the vessels have to stay close to each other, while maintaining at least a certain safety distance. This implies that the vessels need to know where the other vessels are and that there should be protocols for making decisions.

In control theory three main structures exist: centralized, decentralized, and distributed control. Choosing how much to centralize or distribute a system is a core design aspect according to (McCarthy (2014)). In the following paragraphs first the control structure of a single vessel is considered. Thereafter the basic approaches of formation control will be discussed. This section concludes with examples and (dis)advantages for centralized, decentralized and distributed control. Afterwards, in Table 2.2, an overview is presented with all research works that are discussed in this section.

Control structure single vessel Independently from the control structure of the formation, two different approaches to control a single vessel in the formation exist. The first approach is the integral approach, where based on one optimization or calculation the control inputs for the vessel are determined. This approach is used in (Chen et al. (2019c)). The second approach is the modular approach, where two optimizations or calculations take place: in the first optimization or calculation a desired trajectory is determined, in the second optimization or calculation the control inputs necessary to follow that trajectory are determined. This approach is more common and used in (Alonso-Mora et al. (2018); Guo et al. (2019); Pang et al. (2019); Peng et al. (2019); Riahifard et al. (2019)). In Figure 2.3 schematic overviews of the two approaches are given. The main difference between the two approaches is clearly visible: the integral approach computes the control input in one step, whereas the modular approach requires two steps. In this example, a collision-free trajectory is determined based on the position and velocity of the own vessel, the positions of other vessels and obstacles, and a predetermined desired trajectory.

The main advantage of the integral approach is the high performance (Fossen (2011)). On the other hand the main advantage of the modular approach is obviously the modularity that expresses itself mostly in enabling (software) updates for parts of the system (Fossen (2011)) and the ease of problem solving. Furthermore the computation time of the modular approach is expected to be lower in the case of optimization functions, since the size of the optimization problem in the integral approach will be larger. Finally, (Fossen (2011)) states that from an industrial point of view it is attractive to use a modular approach.

Approaches of formation control According to (Schoerling et al. (2010)) three approaches of formation control exist: Behavior-based, virtual structures, and leader-follower approaches. These approaches have been described in (Lawton et al. (2003)) as follows. In the first approach, several desired behaviors are prescribed for each vehicle. The relative importance of each behavior is weighted to create the final control. In the virtual structure approach, the formation is handled as a single entity. The desired motion is assigned to the virtual structure, which makes trajectories for the participants of the formation. In the leader-follower approach, a designated leader is followed by the followers respecting user-defined distances between the leader and follower.

These three approaches can all work in centralized, decentralized, or distributed control architecture. A schematic overview of centralized, decentralized and distributed control is shown in Figure 2.4. These three

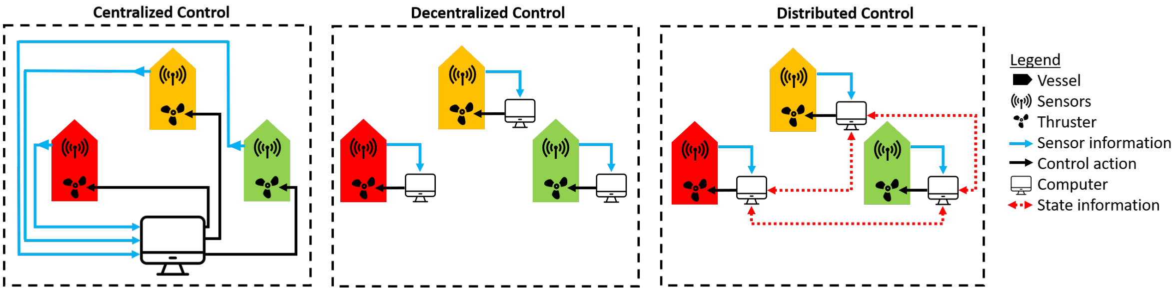


Figure 2.4: Schematic overview of the three main control structures: centralized, decentralized and distributed control

architectures are discussed in detail in the next paragraphs.

Centralized control In a centralized control system one agent, or central controller makes all the decisions for the vehicles. Research works that use centralized control for formation control are among others (Antonelli et al. (2009); Ghommem et al. (2007); Mas and Kitts (2014); Schoerling et al. (2010); Yu et al. (2013)). In (Chen et al. (2018)) both a centralized and distributed controller have been investigated to address the formation control problem. The main advantage of the centralized controller turned out to be that at the same time all information of the different vehicles is known. The consequence is that the centralized controller will make a fair decision that is optimal for the whole system. However, the main issue with centralized control is that it does not scale well, as is shown in several papers (Chen et al. (2018); Kuwata and How (2011); Maestre and Negenborn (2014); Negenborn and Maestre (2014)). The computation time is significantly increased with the number of decision variables, state variables and measurements (Negenborn and Maestre (2014)).

Decentralized control In a decentralized control system, the vehicles make decisions on their own, but do not communicate with each other. This leads to a low computation time, but since no accurate information of the other vehicles is known, the solution is expected to be less accurate compared to a centralized or distributed control system. Research works on decentralized control are (Breivik et al. (2008); Lawton et al. (2003)).

Distributed control In a distributed control system the vehicles communicate with each other and make decisions on their own based on both information of their own sensors and information from other vehicles. Distributed control is widely used in formation control, since it is easily scalable. A downside of distributed control is that decisions have to be made without having all information. Besides, if two vehicles make a decision at the same time, those decisions could have an impact on each other that is note expected since the vehicles have no influence on the decision of others. Distributed control is preferred over centralized control among others in (Alonso-Mora et al. (2017, 2018, 2015); Chen et al. (2018); Kuwata and How (2011); Zhu et al. (2019)) for formation control. In distributed control, multiple options exist. These will be discussed in the following paragraphs.

Multi-layer control In some cases using distributed control, particular vehicles have authority over other vehicles in the decision structure. This is called a multi-layer control structure. This could be applied in a leader-follower strategy, where the leader determines the trajectory, but the followers are able to determine their own route based on the information of the leader, the other vehicles and their own sensors.

Iterative or non-iterative distributed control The trajectory optimization problem in the work of (Kuwata and How (2011)) is solved with a non-iterative strategy. Every vehicle solves its own problem in sequence, while in the same time generating feasible modifications to the prediction of the plan of other vehicles. This leads to a low amount of communication as well as computation time. The overall objective can be improved by the sacrifice of objectives of some of the vehicles.

On the contrary in an iterative framework, agents obtain agreement after exchanging information about their own decision and the preferences of other vehicles. According to (Negenborn and Maestre (2014)) iterative frameworks have a larger potential to achieve overall optimal performance.

Parallel or serial In the distributed control strategies with parallel schemes, all the agents perform computations at the same time. Due to potential conflicts of objectives, the solutions may be not converged. This problem is sometimes solved by adding a hierarchical layer to the control structure (Chen et al. (2018)). The advantage of serial schemes, where only one agent is performing computations at a time, is that the agents possess the most up-to-date information of the other agents. The research in (Negenborn and Maestre (2014)) shows that the serial scheme has better results than the parallel scheme in terms of solution speed, by requiring fewer iterations, and solution quality.

In (Chen et al. (2018)) it appears that different updating sequences have a big role in the outcome of the serial scheme. If two vessels approach each other and they have to dodge each other to prevent a collision, the vessel that is the first in the sequence will always sacrifice more than the other. This is because at first, the planned trajectory of both vessels is unchanged. After the first vessel has changed its trajectory, the danger is gone, or at least less severe. The first vessel that has to change its trajectory will therefore do more to avoid a collision than the second vessel.

Table 2.2: Control architectures used in formation control of autonomous vehicles

Sector	**Type of vehicle**	**References**	**Formation control approach**[a]	**Control structure**[b]	**Control approach**[c]	**Validated by**[d]
Ground vehicles	Mobile robots	Lawton et al. (2003)	BB	DC	M	SE
	Mobile robots	Antonelli et al. (2009)	BB	C	M	SE
	Mobile manipulators	Alonso-Mora et al. (2017)	GT	C	M	SE
Aerial vehicles	Quadrotors	Kuwata and How (2011)	-	D:S-NI-S	I	SE
	Quadrotors	Yu et al. (2013)	GT	C	I	SE
	Quadrotors	Alonso-Mora et al. (2015)	-	D:S-NI-P	M	SE
	Quadrotors	Alonso-Mora et al. (2018)	GT	D:S-I-S	M	SE
	Quadrotors	Zhu et al. (2019)	GT	D:S-I-P	M	SE
	Quadrotors	Guo et al. (2019)	GT	D:S-NI-P	M	SE
Surface vehicles	Vessel	Ghommem et al. (2007)	VS	C	M	S
	Vessel	Breivik et al. (2008)	LF	DC	M	S
	Vessel	Schoerling et al. (2010)	LF	C	M	SE
	Vessel	Mas and Kitts (2014)	CS	C	I	SE
	Vessel	Chen et al. (2018)	VTF	D:S-I-S	I	S
	Vessel	Chen et al. (2019c)	VTF	D:S-I-S	I	S
	Vessel	Riahifard et al. (2019)	LF	C	M	S
	Vessel	Peng et al. (2019)	PG	D:S-NI-P	M	S
Underwater vehicles	Towfish	Pang et al. (2019)	GT	D:S-NI-S	M	S

[a] Leader-Follower (LF), Graph Theory (GT), Virtual Structure (VS), Path Guided (PG), Behavior-based (BB), Cluster Space (CS), VTF (attempt to stay close to nearby vessels), No formation control (-)
[b] Centralized (C), Decentralized (DC), Distributed (D): Single (S) / Multi (M) layer, Iterative (I) / Non-iterative (NI), Parallel (P) or serial (S)
[c] Modular (M), Integral (I)
[d] Simulation (S), Experiment (E)

2.1.3. Control method

Another important characteristic of a formation control system, is the control method. This is the part of the system that translates the desired input value (in most cases a desired trajectory) to an action. In this section, widely used control methods will be discussed. An overview of the discussed research works will be given in Table 2.3, placed in the end of this section.

Model Predictive Control Since its early days model predictive control (MPC) has been popular in practical applications (Mayne (2014)). Based on the model of the vehicle the MPC controller determines the future reactions on different inputs and determines the most optimal sequence of actions for the vehicle. After the first action in the sequence is applied, the optimization problem is solved again, to create a new sequence based on new information. According to (Chen et al. (2018)) MPC has many advantages for waterborne transport. First of all the manoeuvrability of vessels is poor and the predictive property of MPC is able to detect conflicts in an early state. Second, MPC considers the latest available information, which is beneficial to deal with the uncertainties of the trajectory of a vessel due to external disturbances such as wind, wave and current.

In (Abdelaal et al. (2016)) nonlinear model predictive control (NMPC) is used for trajectory tracking and collision avoidance. The maximum execution time in simulation environment in that research was 16 ms, which is low enough to use it real-time. However Chen et al. (2019c) has linearized the dynamic model to design the MPC algorithm that was used to create a cooperative vessel train formation, because it is expected that an NMPC algorithm would be too time consuming to use in real-time applications. If the NMPC algorithm is fast enough to use in real-time application, it is is preferred over the linearized MPC since the NMPC will have more reliable results as the model used resembles the dynamics of the vessel better. MPC is also used in (Alonso-Mora et al. (2018); Guo et al. (2019); Perizzato et al. (2015); Wang et al. (2019); Zheng et al. (2019); Zhu et al. (2019)).

PID control A widely used control method that is less complicated to implement is Proportional-Integral-Derivative (PID) control. A PID controller calculates the error between the reference value and the measured value. Based on proportional, integral and derivative terms of the error, the PID controller determines the action. The PID controller does not predict future steps. Therefore the PID controller does not require the availability of a model and has a low computation time. PID control is used in (Liu et al. (2020); Manoharan and Chiu (2019); Mastellone et al. (2008); Schoerling et al. (2010))

Feedback linearization According to (Fossen (2011)) the main idea of Feedback linearization is to transform the nonlinear open-loop system dynamics into linear closed-loop system dynamics. Subsequently other control methods can be applied to the linear system. Since the computation time necessary to optimize a linear system is lower compared to a nonlinear system, Feedback linearization could be beneficial to use. Unlike other linearization methods, Feedback linearization does not linearize the system dynamics, but chooses the input force in such a way that the non-linearities of the system are cancelled out. Feedback linearization is used in the work of (Haseltalab and Negenborn (2019)).

(Integrator) Backstepping The aforementioned feedback linearization approach leads to cancellation of useful nonlinearities in most cases. More flexibility is given in the Backstepping approach, since it does not require the resulting input-output dynamics to be linear. The main idea of Backstepping is to use state variables as virtual controls and depending on the dynamics of each state, design intermediate control laws. The Backstepping approach is a recursive procedure - hence the name - where a Lyapunov function is derived for the entire system. When the system is expanded with an integrator, the approach is called Integrator Backstepping. A very extensive explanation on Integrator backstepping is given in (Fossen (2011)). This method is used in (Breivik et al. (2008); Ghommem et al. (2007)).

Sliding mode control Sliding mode control is a robust nonlinear control method, which incorporates techniques to handle model uncertainty (Fossen (2011)). A main advantage of Sliding mode control is that it can switch from one control structure to another, based on the current position in the state space. This control method is extensively explained in (Fossen (2011)), where also use cases can be found. Sliding mode control is used in (Schoerling et al. (2010)).

Null-Space-Based The Null-Space-Based (NSB) method decomposes the overall task into several individual control subtasks and establishes a task function for each subtask to ensure that each function can complete the corresponding control target. Subsequently, the subtasks are divided into different priorities. The NSB method uses a geometric hierarchy-based composition of the outputs of the tasks to obtain the commands for the low level controllers of the vehicle. This method is used in (Antonelli et al. (2009); Pang et al. (2019)).

Table 2.3: Control methods used in formation control of autonomous vehicles

Sector	Type of vehicle	References	Control Method[a]	Validated by[b]
Ground vehicles	Mobile robots	Lawton et al. (2003)	PD	SE
	Segbot	Mastellone et al. (2008)	P	SE
	Mobile robots	Antonelli et al. (2009)	PID	SE
	Unicycle vehicles	Perizzato et al. (2015)	MPC	SE
	Mobile manipulators	Alonso-Mora et al. (2017)	LQR	SE
	Cars	Zheng et al. (2019)	MPC	S
	Cars	Wang et al. (2019)	MPC	S
	Differential drive robots	Manoharan and Chiu (2019)	P	S
Aerial vehicles	Quadrotors	Kuwata and How (2011)	MPC	SE
	Quadrotors	Yu et al. (2013)	PD	SE
	Quadrotors	Alonso-Mora et al. (2015)	LQR	SE
	Quadrotors	Alonso-Mora et al. (2018)	NMPC	SE
	Quadrotors	Zhu et al. (2019)	NMPC	SE
	Quadrotors	Guo et al. (2019)	MPC	SE
	Tailsitter	Liu et al. (2020)	PD	S
Surface vehicles	Vessel	Ghommem et al. (2007)	LDM & BS	S
	Vessel	Breivik et al. (2008)	IB	S
	Vessel	Schoerling et al. (2010)	SM & PID	SE
	Vessel	Mas and Kitts (2014)	MB	SE
	Vessel	Abdelaal et al. (2016)	NMPC	S
	Vessel	Chen et al. (2018)	MPC	S
	Vessel	Chen et al. (2019c)	MPC	S
	Vessel	Riahifard et al. (2019)	NN	S
	Vessel	Peng et al. (2019)	MB	S
	Vessel	Haseltalab et al. (2019)	FL MPC	S
Underwater vehicles	Towfish	Pang et al. (2019)	NSB	S

[a] PID, MPC, Linear-Quadratic Regulator (LQR), Lyapunov Direct Method (LDM), Backstepping (BS), Integrator Backstepping (IB), Sliding Mode (SM), Model-Based (MB), Neural Network (NN), Feedback-Linearization (FL), Null-Space-Based (NSB)
[b] Simulation (S), Experiment (E)

2.1.4. Formation control in aerial and ground applications

Formation control is a trending topic in the trucking industry, where formation control - or platooning - of trucks could reduce the overall fuel consumption up to 20% (Liang et al. (2016)). Moreover, many researchers have investigated the concept of formation control using mobile robots. Useful applications could be among others transporting big objects, performing field research or conducting military operations. The main contributions to the field of formation control in ground applications have been described in (Alonso-Mora et al. (2017); Antonelli et al. (2009); Lawton et al. (2003); Manoharan and Chiu (2019); Mastellone et al. (2008); Perizzato et al. (2015); Wang et al. (2019); Zheng et al. (2019)).

Within the aerial industry, formation control has also been subject of research. Drivers for this research include fuel savings (Zhang and Liu (2019)) and conducting military operations (Yu et al. (2013)). The main contributions to the field of formation control in aerial applications have been described in (Alonso-Mora et al. (2018, 2015); Dong et al. (2019); Guo et al. (2019); Kuwata and How (2011); Liu et al. (2020); Yu et al. (2013); Zhu et al. (2019)).

Examples of usage of (autonomous) formation control in various applications are shown in Figure 2.5. Compared to the other sectors, maritime formation control is relatively young and less experiments have been performed in this sector as is visible in the tables throughout this chapter. The most relevant observations that can be made from the literature on formation control in aerial and ground applications compared to the maritime formation control are the following:

- Many research works on aerial and ground applications have been validated by real experiments;

(a) Truck platooning ©2014 Scania CV AB (b) Maritime formation ©2010 U.S. Navy (c) Drones in formation ©2019 New China TV

Figure 2.5: Application of formation control in different sectors

- Predictive control methods have been validated in experiments with aerial and ground vehicles, while most maritime literature state that these control methods result in too high computation times;
- The graph theory formation control approach is frequently used in other applications than maritime;
- The motivation for formation control in aerial and ground applications is often the saving on fuel consumption, while in maritime formation control this is only mentioned in (Chen et al. (2019a)). The main motivation for maritime formation control is efficiency for applications such as search and rescue or ocean sampling, by avoiding congestions, and by passing bridges and locks faster.

These observations are relevant, since they point out possible research areas for maritime formation control. This will be discussed in more detail in the next section.

2.2. Research gaps in maritime formation control

As pointed out in the previous section and indicated by the various tables throughout this chapter (Table 2.1, Table 2.2, and Table 2.3), real experiments regarding formation control have barely been performed with autonomous surface vessels (ASVs). This impression is confirmed in the extensive literature overview in (Liu et al. (2016)) that states that in only five research works, real experiments have been performed regarding formation control with ASVs. In a more recently published extensive literature review on leader-follower formation control (Rashid et al. (2019)) no extra works on ASVs that performed experiments were mentioned. All research works on maritime formation control that have been validated by real experiments that have been found, are listed in Table 2.4.

Table 2.4: Research works on maritime formation control validated by real experiments

References	**# of ASVs used**	**Formation control approach**[a]	**Control structure**[b]	**Control method**[c]	**Collision avoidance included?**
Kyrkjebø et al. (2007)	1[d]	LF	C	MB, PD	No
Mahacek et al. (2009)	3	CS	C	PID	No
Arrichiello et al. (2010)	2	BB	D:S-NI-P	PI	Yes
Schoerling et al. (2010)	1[d]	LF	C	SM, PID	No
Mas and Kitts (2014)	3	CS	C	MB	No
Zhang et al. (2019)	4	LF	C	Fuzzy PID	No
Li et al. (2020)	3	GT	DC	FLMB	No

[a] Leader-Follower (LF), Graph Theory (GT), Behavior-based (BB), Cluster Space (CS)
[b] Centralized (C), Decentralized (DC), Distributed (D): Single (S) / Multi (M) layer, Iterative (I) / Non-iterative (NI), Parallel (P) or serial (S)
[c] PID, Sliding Mode (SM), Model-Based (MB), Feedback-Linearization Model-Based (FLMB)
[d] Plus one virtual vessel

From Table 2.4, it can be deduced that the research works validated with real experiments do not use virtual structure as formation control approach. Furthermore the control structure was in almost all cases centralized. Besides, while in the research works validated with simulations many advanced control methods such as MPC and neural networks were used (see Table 2.3), these approaches were not used in the research works validated with real experiments. Finally, it appears that only in one research collision avoidance with other vessels or objects has been handled (Arrichiello et al. (2010)).

A commonly used reason that no predictive controllers have been used in real experiments to research maritime formation control (to the best knowledge of the author), is the computation time. For example (Li et al. (2020)) states that "Model predictive control is a method with strong theoretical basis and massive computation, so that it is very difficult to implement, and not applicable to practical engineering." As discussed in Section 2.1.4 the predictive methods however have been used in experiments with aerial and ground vehicles. This seems to be a major research gap.

Furthermore it is stated in many of the research works that there is great demand for experimental data at all (Liu et al. (2016); Rashid et al. (2019); Zhang et al. (2019)). Other research gaps, pointed out in the previous sector, are 1) the usage of graph theory formation control approach, and 2) the amount of savings in fuel consumption when sailing in formation.

2.3. Conclusion

Throughout this chapter, answers have been given on the research question '*What is the current state of research on formation control?*'. Formation control has been mainly researched in four sectors: autonomous aerial vehicles, autonomous ground vehicles, autonomous surface vehicles, and autonomous underwater vehicles. Research works on formation control use either fixed or flexible configurations, depending on the goal of the formation. Moreover, in general three approaches for formation control exist: behavior-based, virtual structures, and leader-follower approaches. Independent of the formation control approaches, three control architectures can be distinguished: centralized, decentralized, and distributed control. The scientific world points out that a distributed approach has the most potential for the application of maritime formation control.

In simulation many control methods have been proposed for maritime formation control. However during the few real experiments that have been carried out with vessels, only relatively simple controllers have been used. Predictive control methods have not been tested in maritime formation control experiments, but performed experiments with aerial and ground vehicles indicate the large potential for these control methods. The biggest research gap in maritime formation control is the gathering of experimental data in general, since that data seems to be missing.

In the following chapter, a strategy to control one Roboat fully autonomously will be presented based on the literature review in this chapter. The knowledge gained during the discussed research works will be used as foundation for the remaining parts of this research. Especially in Chapter 4 this will be very useful, since the control of one Roboat is expanded to a formation control strategy in that chapter.

3

Fully autonomous control of a Roboat

In this chapter, the algorithms that have been developed in this research to control a single Roboat fully autonomously are discussed. With that, this chapter provides the first part of the answer of research question *'How to design a flexible formation control strategy?'*. The next chapter discusses the control strategies for multiple Roboat units and will provide the second and last part of the answer of the aforementioned research question.

The following relevant parts of the literature review performed in Chapter 2 that are applicable to the control of a single Roboat will be taken into account. First of all the study on the control structure of a single vessel in Section 2.1.2 provides useful information. Secondly, the discussion on the different control methods, discussed in Section 2.1.3, is relevant. Finally, the research gaps that have been pointed out in Section 2.2 will be filled in.

This chapter is organized as follows. First the control structure is explained. Then, one section is dedicated per part of the control structure: marine craft, navigation system, guidance system, and control system.

3.1. Control structure

As discussed in Section 2.1.2, in general two approaches for the control structure of a single vessel are used in literature: 1) the integral approach, and 2) the modular approach. As shown in that section, the majority of the research works use the modular approach. The main advantage of the integral approach is the high performance (Fossen (2011)). On the other hand the main advantage of the modular approach is obviously the modularity that expresses itself mostly in enabling (software) updates for parts of the system (Fossen (2011)) and the ease of problem solving. Furthermore the computation time of the modular approach is expected to be lower in the case of optimization functions, since the size of the optimization problem in the integral approach will be larger. Finally, (Fossen (2011)) states that from an industrial point of view it is attractive to use a modular approach.

While the results of the integral approach are expected to be (slightly) better than the results of the modular approach, the latter has appealing practical advantages. Therefore in this research the commonly used modular approach will be followed.

In (Fossen (2011)) a widely used, general modular control structure is discussed: the guidance, navigation and control (GNC) structure. The control structure that will be used for the Roboat units is deduced from this approach and schematically shown in Figure 3.1. The structure consists of four subsystems:

1. **Marine craft**: The vessel itself, sensors gather information and control actions are performed by the actuator(s);
2. **Navigation system**: Determines the position and velocity of the vessel;
3. **Guidance system**: Responsible for the computation of the reference trajectory;
4. **Control system**: Determines the correct input to the actuators to make the vessel follow the reference trajectory.

The following sections will address the design of these four subsystems.

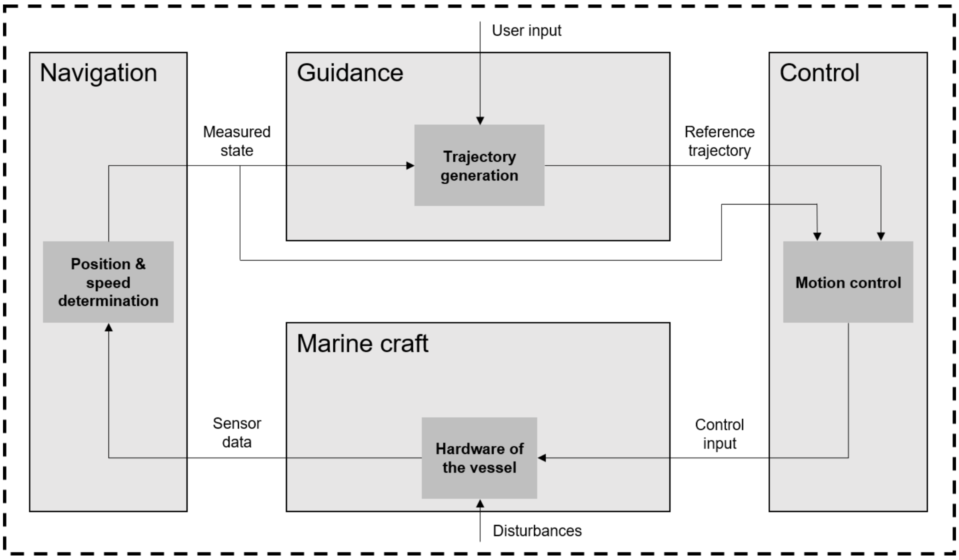
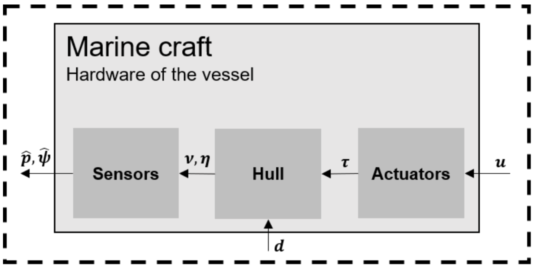


Figure 3.1: General control system of a vessel according to Fossen (2011)



Declaration of symbols		
$\boldsymbol{u}$	$\in \mathbb{R}^{n_a \text{x} 1}$	Control input ($n_a \in \mathbb{N}$; # of actuators)
$\boldsymbol{\tau}$	$\in \mathbb{R}^{3\text{x}1}$	Internal forces and moments
$\boldsymbol{d}$	$\in \mathbb{R}^{3\text{x}1}$	External forces and moments
$\boldsymbol{\eta}$	$\in \mathbb{R}^{3\text{x}1}$	Position
$\boldsymbol{\nu}$	$\in \mathbb{R}^{3\text{x}1}$	Velocity
$\hat{\boldsymbol{p}}$	$\in \mathbb{R}^{2\text{x}1}$	Measured position without heading
$\hat{\boldsymbol{\psi}}$	$\in \mathbb{R}$	Measured heading

Figure 3.2: Detail of the control system of Roboat: Marine craft

3.2. Marine craft

Perhaps the most trivial part of the general control of a vessel is the marine craft itself. This is the part where both information is gathered via onboard sensors and the proposed control strategy is executed by sending the control input to the actuator(s). An overview of the marine craft is schematically shown in Figure 3.2. As shown, the input of the marine craft is the control input vector. This vector can contain the amount of force or power per thruster that is required. In case of a rudder or rotatable thrusters, this vector also contains required angles or other inputs that influence the angle of the rudder or rotatable thruster. The output of the marine craft is the measured position.

During simulations, the behaviour of the marine craft - in this research the Roboat - must be simulated. In the research of (Wang et al. (2018)) a model of the 1:4 scale Roboat is assumed and parameters have been determined. In this research that model and parameters will be used, since the same set-up of the 1:4 scale Roboat units will be used and the results of their research were satisfying. The model is deduced from the widely accepted model suggested by (Fossen (2011)). The dynamics of a vessel can be generically described by the following nonlinear differential equation:

$$\mathbf{M}\dot{\nu} + \mathbf{C}(\nu)\nu + \mathbf{D}\nu = \tau + \mathbf{d} \tag{3.1}$$

where $\nu = [\epsilon, \upsilon, r]^{\mathrm{T}}$ denotes the velocity, containing the surge velocity (ϵ), sway velocity (υ), and yaw rate (r) in the body fixed frame; $\mathbf{M} \in \mathbb{R}^{3x3}$ is the added mass and inertia matrix; $\mathbf{C}(\nu) \in \mathbb{R}^{3x3}$ is the matrix of Coriolis and centripetal terms; $\mathbf{D} \in \mathbb{R}^{3x3}$ is the drag matrix; $\tau \in \mathbb{R}^{3x1}$ is the vector of body-frame internal forces and moments applied to the vessel; and $\mathbf{d} \in \mathbb{R}^{3x1}$ the vector of body-frame external forces and moments applied to the vessel, for instance due to wind and waves. The position and orientation of the vessel in the North-East-Down (NED) coordinate system is defined as $\eta = [x, y, \psi]^{\mathrm{T}}$. This is graphically displayed in Figure 3.3.

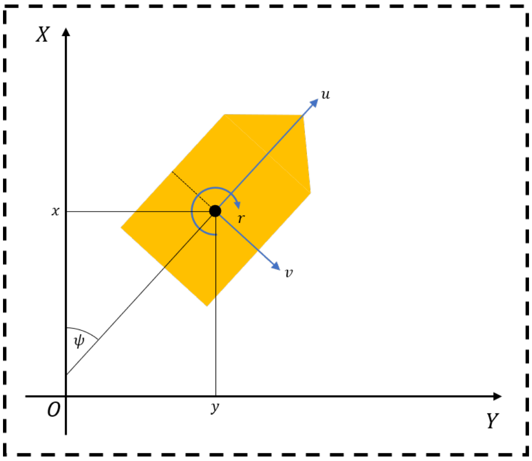


Figure 3.3: The position of the vessel is expressed in the NED frame, while the velocity (blue) is expressed in the body fixed frame

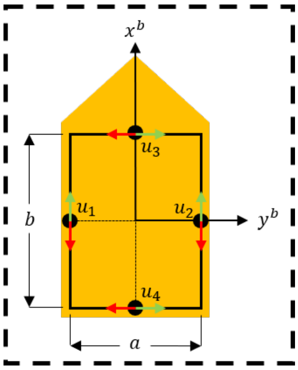


Figure 3.4: Roboat's thrust configuration. Each thruster is fixed and can generate positive (green) and negative (red) forces

The kinematic equation relating the velocity components in the NED frame to those in the body fixed frame is:

$$\dot{\eta} = \mathbf{R}(\psi)\nu \tag{3.2}$$

where $\mathbf{R}(\psi)$ is the rotation matrix converting a position or velocity vector from the body fixed frame to the inertial frame:

$$\mathbf{R}(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Using the same convention as (Fossen (2011)), it will be indicated when a position or velocity is described relative to a coordinate system other than the above specified coordinate system for the concerned state, with a superscript. This means that a position relative to the body-fixed frame will be described as $\eta^b = [x^b, y^b, \psi^b]^{\mathrm{T}}$, and a velocity relative to the NED frame will be described as $\nu^n = [\epsilon^n, \upsilon^n, r^n]^{\mathrm{T}}$.

Considering that in the specific case of the 1:4 scale Roboat the origin of the body-fixed frame coincides with the center of mass of the vessel and that the vessel is moving at low speeds, the mass matrix, matrix of Coriolis and centripetal terms and the drag matrix can be described as follows (Wang et al. (2018)):

$$\mathbf{M} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix} \tag{3.4}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & -m_{22}\upsilon \\ 0 & 0 & m_{11}\epsilon \\ m_{22}\upsilon & -m_{11}\epsilon & 0 \end{bmatrix} \tag{3.5}$$

$$\mathbf{D} = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_\upsilon & 0 \\ 0 & 0 & N_r \end{bmatrix} \tag{3.6}$$

The parameters that are used in this research, have been determined experimentally in the research of (Wang et al. (2018)) and are shown in Table 3.1.

Table 3.1: Results of the parameter identification conducted by Wang et al. (2018).

Item	m_{11}	m_{22}	m_{33}	X_u	Y_υ	N_r
Value	12.982	23.318	1.273	6.012	7.112	0.771

The thrust configuration of the Roboat is graphically shown in Figure 3.4 and since the thrust configuration of the Roboat is known, τ can be written as function of the forces produced per thruster:

$$\tau = \mathbf{B}\mathbf{u} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ a/2 & -a/2 & b/2 & -b/2 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} \tag{3.7}$$

where **B** is the thruster allocation matrix and **u** is the control vector, that is the collection of forces generated by the propellers: u_1, u_2, u_3, and u_4. The distance between the transverse propellers is represented by a and the distance between the longitudinal propellers by b. In case of the 1:4 scale Roboat that is used in this research, the parameters are as follows: $a = 0.90$ [m], $b = 0.45$ [m]. Each propeller is fixed and can generate forward and backward forces.

Discretization The dynamic model of a vessel, described in Equation 3.1, is continuous. However since it takes time to calculate the required control actions, measure the state of the vessel and determine a reference trajectory, a discrete model is required. To obtain this discrete model, the same method as proposed in (Chen et al. (2019a); Zheng et al. (2016)) is used. With state vector $\mathbf{x} = \begin{bmatrix}\eta^{\mathrm{T}} & \nu^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, the dynamic model (3.1)-(3.2) can be expressed as:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \tau, \mathbf{d}) \\ &= \begin{bmatrix} \mathbf{0}^{3\times3} & \mathbf{R}(\psi) \\ \mathbf{0}^{3\times3} & \mathbf{M}^{-1}(-\mathbf{C}(\nu) - \mathbf{D}) \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0}^{3\times3} \\ \mathbf{M}^{-1} \end{bmatrix} \tau + \begin{bmatrix} \mathbf{0}^{3\times3} \\ \mathbf{M}^{-1} \end{bmatrix} \mathbf{d} \end{aligned} \tag{3.8}$$

The dynamic model (3.8) is now discretized with sampling time t_s to calculate the state at time step $k+1$:

$$\mathbf{x}(k+1|k) = \mathbf{x}(k) + \int_{k t_s}^{(k+1)t_s} f(\mathbf{x}, \tau, \mathbf{d})\, dt \tag{3.9}$$

The discrete differential equation (3.9) is solved every time step k using the ode45 solver in MATLAB.

Output of the system Finally, the output of the marine craft, is the information of the sensors. In simulation, this will be the exact position and velocity of the 1:4 scale Roboat that are coming from the discrete differential equation (3.9). During the implementation, the position of the vessel will be measured using an RTK GPS unit, while the heading will be measured with an Inertial Measurement Unit (IMU). The rough data will be passed to the navigation system. More details on the implementation can be found in Chapters 5 and 6.

3.3. Navigation system

In general, the goal of the navigation system is to determine the position, attitude, course, distance travelled, velocity and / or acceleration of a vessel (Fossen (2011)), based on the data of the marine craft's sensors. The outcome of the navigation system is input for the control system and guidance system of the vessel. In most research works (among many others Chen et al. (2019b); Peng et al. (2019); Riahifard et al. (2019)) the measured position, heading, and velocity are required as output of the navigation system. Later in this chapter, in Section 3.5, it will become clear that this is also the case for the proposed control system in this research. A schematic overview of the navigation system is shown in Figure 3.5.

In simulation, the position, heading, and the velocity will be a direct output of the marine craft's dynamic model (3.9). To simulate the real behaviour of the sensors, a noise signal can be added to the signal. In the simulations that are included in the results of this thesis, no noise signals are added.

During the real experiments that are performed in this research, the position of the Roboat is determined using an onboard RTK GPS sensor, while the heading is determined using an IMU. Details about the set-up during the experiments can be found in Chapter 6. Using consecutive measurements of the position and heading, the velocity is approximated as follows:

$$\hat{\nu}(k) = \mathbf{R}^{-1}(\overline{\psi}(k)) * \frac{\hat{\eta}(k) - \hat{\eta}(k-1)}{t_s} \tag{3.10}$$

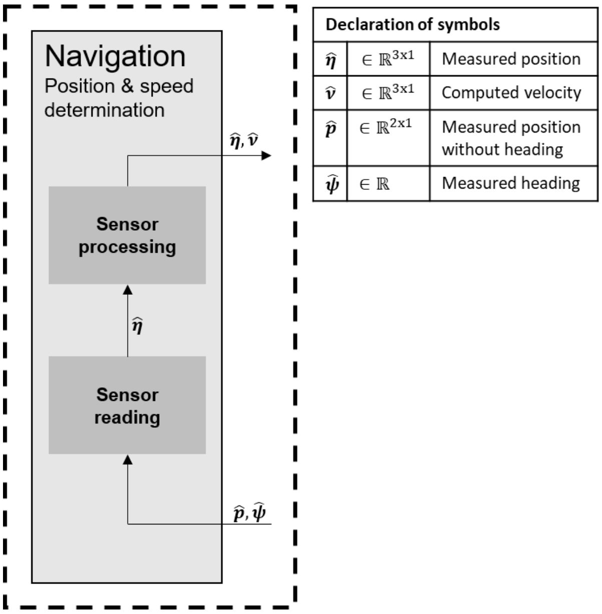


Figure 3.5: Detail of the control system of Roboat: Navigation systems

where

$$\overline{\psi} = \frac{\hat{\psi}(k) + \hat{\psi}(k-1)}{2} \tag{3.11}$$

with k being the current time step, t_s the sampling time, $\hat{\psi} \in \mathbb{R}$ the measured heading, $\hat{\eta} \in \mathbb{R}^{3\times1}$ the measured position, and $\hat{v} \in \mathbb{R}^{3\times1}$ the computed velocity.

An advanced feature of the navigation system that has not been discussed so far, is the ability to detect and predict the (future) positions of other vessels, canal boundaries and obstacles. In order to exploit this advanced feature, sensors that are able to detect objects must be mounted on the vessel. Sensors that can be used are among others: Lidar, Radar, Distance sensor, and Depth camera. For this research the design of an algorithm that is able to detect objects is out of scope. However to include the ability of avoiding objects, vessels and canal boundaries will be taken into account by adding positions of predetermined virtual static objects directly into the guidance system.

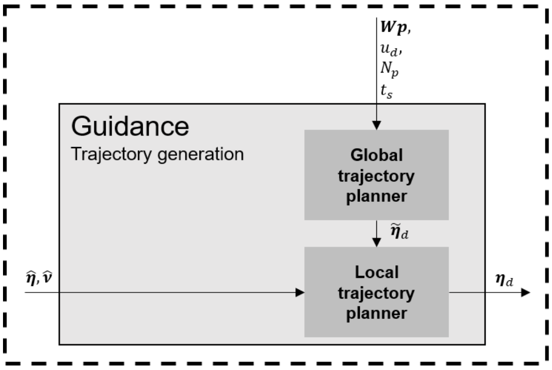
3.4. Guidance system

Every time step, the guidance system computes the reference (desired) position, velocity and/or acceleration of a vessel to be used by the motion control system (Fossen (2011)). In literature different approaches to accomplish this can be found. A distinction can be made in the following categories according to Fossen (2011):

- **Setpoint regulation**: the desired position is chosen to be constant;
- **Trajectory tracking**: the objective is to force the system output $\eta(t)$ to track a desired output $\eta_d(t)$;
- **Path following**: the objective is to follow a predefined path independent of time.

While trajectory tracking is the most complicated approach for the guidance system, since it also takes time in consideration, it is the most promising to use for the application of formation control. For example a leader-follower approach could be achieved using trajectory tracking for the follower with as input the recent positions of the leader. In Figure 3.6 a schematic overview of the guidance system of Roboat without communication with other vessels is shown.

A guidance system can use manual input, external inputs, earth topological information, obstacle and collision avoidance data and the state vector (Fossen (2011)). The manual input can be a predefined trajectory. The external input may include weather data, such as measured wind, wave and current speeds and directions. Earth topological data could be a digital chart. The obstacle and collision avoidance data can be



Declaration of symbols		
$\hat{\boldsymbol{\eta}}$	$\in \mathbb{R}^{3\text{x}1}$	Measured position
$\hat{\boldsymbol{\nu}}$	$\in \mathbb{R}^{3\text{x}1}$	Computed velocity
$\boldsymbol{Wp}$	$\in \mathbb{R}^{2\text{x}n_{Wp}}$	Waypoints ($n_{Wp} \in \mathbb{N}$; # of waypoints)
u_d	$\in \mathbb{R}$	Reference surge speed
N_p	$\in \mathbb{N}$	Number of prediction steps
$\tilde{\boldsymbol{\eta}}_d$	$\in \mathbb{R}^{3\text{x}(N_p+1)}$	Global reference trajectory
$\boldsymbol{\eta}_d$	$\in \mathbb{R}^{3\text{x}(N_p+1)}$	Reference trajectory
t_s	$\in \mathbb{R}_{>0}$	Sampling time

Figure 3.6: Detail of the control system of Roboat: Guidance system without communication

seen as positions of obstacles and other vessels. Finally the state vector consists of the position and speed of the own vessel.

For use in the area of Amsterdam's canals, the guidance system must be able to produce a collision free route through the navigable area. This goal and the availability of sensors - see Chapter 5 - are taken into account in determining the final operation of the guidance system and its corresponding inputs. Hereafter, a method is proposed and the required inputs of the system are discussed as well.

A systematic method to generate a trajectory, while including static and dynamic constraints is to use an optimization method. A global trajectory will be considered to be predefined and the goal of the optimization method is to minimize the difference between the global trajectory and the generated "local" trajectory, while ensuring that the trajectory is collision free and in the navigable area. In the following two sections, the creation of the global and local trajectory is discussed.

3.4.1. Global trajectory planner

In order to create a navigable trajectory, first a global trajectory is created. The inputs of the global trajectory planner are:

1. Waypoints between which straight lines can be drawn;
2. The desired surge speed of the vessel;
3. The sampling time.

This is also shown in Figure 3.6. The global trajectory planner connects the waypoints with straight lines while smoothing the curves in the trajectory, ensuring that the vessel does not encounter sharp turns. The output of the global trajectory planner is a list with coordinates that need to be reached at a certain time. The output of the global trajectory planner is shown in Figure 3.7.

3.4.2. Local trajectory planner

Using the global reference trajectory as input, the local trajectory planner determines the reference trajectory based on the following optimization problem:

$$J = \min_{\mathbf{p}_1} \sum_{i=0}^{N_p} \left\| \tilde{\mathbf{p}}_1(k+i) - \mathbf{p}_1(k+i) \right\| \tag{3.12}$$

subject to

$$\mathbf{p}_1(k) = \hat{\mathbf{p}}_1(k) \tag{3.13}$$

$$\left\| \mathbf{p}_1(k+i) - \mathbf{p}_j(k+i) \right\| \geq d_{\text{safe}}, \qquad i = 1, \ldots, N_p;\ j = 2, \ldots, m \tag{3.14}$$

$$v_{\min} * t_s \leq \left\| \mathbf{p}_1(k+i) - \mathbf{p}_1(k+i-1) \right\| \leq v_{\max} * t_s, \qquad i = 1, \ldots, N_p \tag{3.15}$$

where $\mathbf{p}_1(i) \in \mathbb{R}^{2\times1}$ is the reference trajectory without heading at time step i; $\tilde{\mathbf{p}}_1(i) \in \mathbb{R}^{2\times1}$ is the global reference trajectory without heading at time step i; $\hat{\mathbf{p}}_1(k) \in \mathbb{R}^{2\times1}$ is the measured position without heading at time step k; $\mathbf{p}_j(i) \in \mathbb{R}^{2\times1}$ is the expected position of object or vessel $j \in \mathbb{N}$, where $j = 1$ is the index for the own

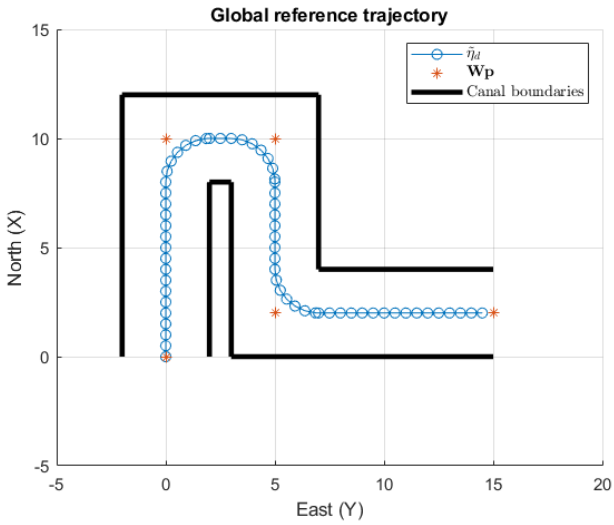


Figure 3.7: Graphically displayed output of the global trajectory planner, with inputs $\epsilon_d = 1$[m/s], $t_s = 0.5$[s]

vessel and m is the total amount of considered objects and vessels; $d_{\text{safe}} \in \mathbb{R}_{>0}$ is the safety distance between the center of gravity of the own vessel and other vessels or objects; $v_{\min} \in \mathbb{R}$ is the minimum speed of the vessel; $v_{\max} \in \mathbb{R}$ is the maximum speed of the vessel; $t_s \in \mathbb{R}_{>0}$ is the sampling time; and $N_p \in \mathbb{N}$ is the number of prediction steps.

The constraints of the optimization problem make sure that:

1. The route is collision free;
2. Successive reference positions are in the reach of the vessel when sailing at maximum speed;
3. The starting point of the local reference trajectory is the current position of the vessel.

This optimization problem is transformed into an MILP problem and solved in MATLAB using the intlinprog solver every time step k. Using the reference trajectory without heading, the reference trajectory is calculated as follows:

$$\psi_d(k+i) = \arctan\left(\frac{y_d(k+i) - y_d(k+i-1)}{x_d(k+i) - x_d(k+i-1)}\right) \qquad i = 1,\ldots,N_p; \tag{3.16}$$

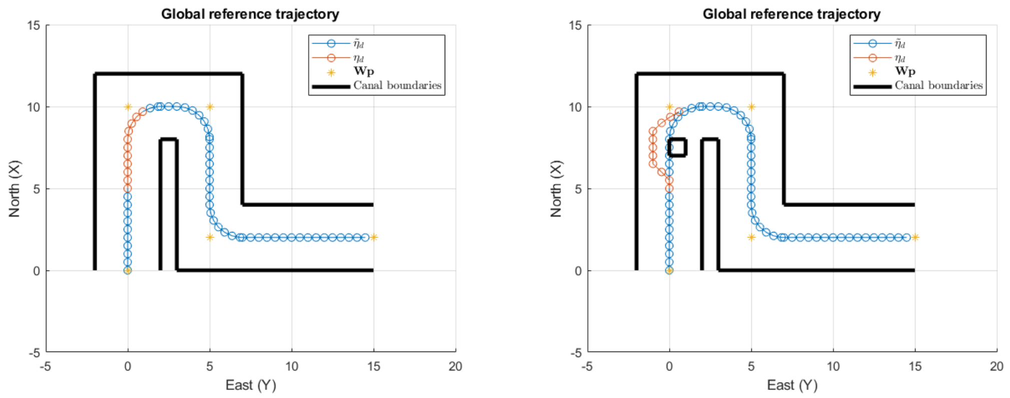
$$\eta_d(k+i) = \begin{pmatrix} \mathbf{p}_1(k+i) \\ \psi_d(k+i) \end{pmatrix} \equiv \begin{pmatrix} x_d(k+i) \\ y_d(k+i) \\ \psi_d(k+i) \end{pmatrix} \qquad i = 1,\ldots,N_p; \tag{3.17}$$

where $\eta_d(i) \in \mathbb{R}^{3\times 1}$ is the reference trajectory at time step i, consisting of the components in x, y, and ψ direction: $x_d(i)$, $y_d(i)$, and $\psi_d(i)$.

The influence of the local trajectory planner is graphically displayed in Figure 3.8. In the figure, the same trajectory as in the example showed in Figure 3.7 is used. The output of the local trajectory planner at time step 11 ($t = 5$[s]) without and with an extra object added are shown. It is evident that the local trajectory planner avoids the object, while maintaining a distance of at least the safety distance to the canal boundaries, and minimizing the error to the global reference trajectory. The canal boundaries and objects have been taken into account by regarding them as a sequence of point masses, where every point mass is added to p_j.

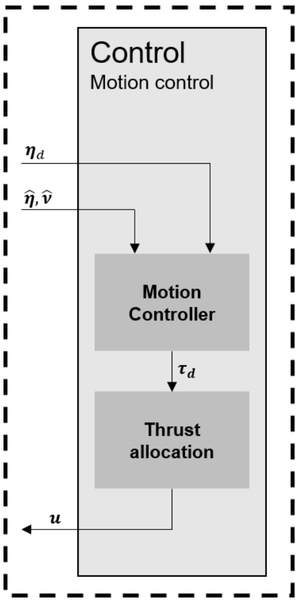
3.5. Control system

Whereas the guidance system creates a navigable trajectory for the vessel, the goal of the control system is to give power to the actuators of the vessel in such a way that the trajectory is followed by the vessel. In most cases, the control system itself can be split in two subsystems: 1) the motion control system, and 2) the thrust allocation (Fossen (2011)). The former determines the necessary forces and moments that are required to act



(a) Without extra object (b) With an extra object added

Figure 3.8: Graphically displayed output of the local trajectory planner at $k = 11$ (or $t = 5$[s]), with inputs $v_{\text{max}} = 1.5$[m/s], $d_{\text{safe}} = 1$[m], $N_p = 10[-]$ and the global reference trajectory of the example in Figure 3.7



Declaration of symbols		
η_d	$\in \mathbb{R}^{3x(N_p+1)}$	Reference trajectory ($N_p \in \mathbb{N}$; # of prediction steps)
$\hat{\eta}$	$\in \mathbb{R}^{3x1}$	Measured position
$\hat{v}$	$\in \mathbb{R}^{3x1}$	Computed velocity
τ_d	$\in \mathbb{R}^{3x1}$	Reference internal forces and moments
u	$\in \mathbb{R}^{n_a x1}$	Control input ($n_a \in \mathbb{N}$; # of actuators)

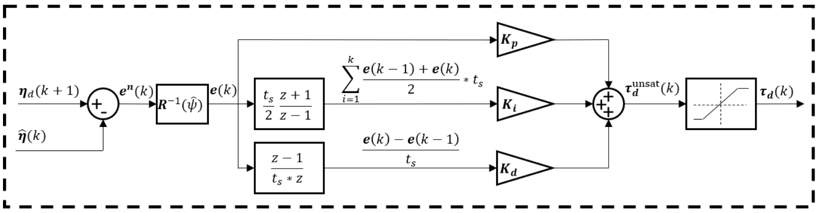
Figure 3.9: Control system of a Roboat unit

on the center of gravity on the vessel to follow the reference trajectory. The thrust allocation subsequently determines the necessary forces per thruster to create the forces and moments acting on the center of gravity. The schematic overview of such a control system is shown in Figure 3.9.

As is discussed in Chapter 2, a frequently used control method during experiments is PID. Research gaps are amongst others the use of predictive control methods during formation control experiments with vessels. The potential of predictive control methods is high, as also pointed out in Chapter 2. In order to design a predictive control method and compare this to existing literature, a baseline PI control method is designed as well as two predictive control methods. All three controllers will be discussed in the following paragraphs.

3.5.1. Option 1: PI control

The discrete PID control method that is used as baseline is shown in Figure 3.10. A multi-input-multi-output PID control algorithm based on the error of the position relative to the reference trajectory is implemented in MATLAB. The PID control algorithm works as follows. Firstly, the error is converted to the body-fixed frame. Then the error is multiplied with tuning parameter $\mathbf{K}_p \in \mathbb{R}^{3\times3}$, the integral of the error is multiplied with tuning parameter $\mathbf{K}_i \in \mathbb{R}^{3\times3}$, and the derivative of the error is multiplied with tuning parameter $\mathbf{K}_d \in \mathbb{R}^{3\times3}$. Subsequently, these three parts are added together and form the unbounded reference internal forces and



Declaration of symbols		
$\boldsymbol{\eta}_d(k)$	$\in \mathbb{R}^{3\times1}$	Reference trajectory at time step k
$\hat{\boldsymbol{\eta}}(k)$	$\in \mathbb{R}^{3\times1}$	Measured position at time step k
$\boldsymbol{e}(k)$	$\in \mathbb{R}^{3\times1}$	Error at time step k
$\boldsymbol{R}(\psi)$	$\in \mathbb{R}^{3\times3}$	Rotation matrix
t_s	$\in \mathbb{R}_{>0}$	Sampling time
$\boldsymbol{K_p}$	$\in \mathbb{R}^{3\times3}$	Proportional gain matrix
$\boldsymbol{K_i}$	$\in \mathbb{R}^{3\times3}$	Integral gain matrix
$\boldsymbol{K_d}$	$\in \mathbb{R}^{3\times3}$	Derivative gain matrix
$\boldsymbol{\tau}_d^{\text{unsat}}(k)$	$\in \mathbb{R}^{3\times1}$	Unbounded reference internal forces and moments
$\boldsymbol{\tau}_d(k)$	$\in \mathbb{R}^{3\times1}$	Reference internal forces and moments

Figure 3.10: PID control method

moments. Finally, this vector is bounded and this forms τ_d. In this research, a PI control method is used, so $\mathbf{K}_d = \mathbf{0}^{3\times3}$.

As can be seen in the figure, the output of the PID control is τ_d. The thrust allocation algorithm used, is discussed below.

Thrust allocation Roboat has four thrusters and three degrees of freedom. This means that the thrust allocation is over-actuated. To address this over-actuated thrust allocation problem, an optimization function is designed to minimize the amount of force and the amount of change per thruster. The algorithm used is the following:

$$J = \min_{\Delta\mathbf{u}} \left(\left(\mathbf{u}(k-1) + \Delta\mathbf{u}(k)\right)^{\mathrm{T}} \mathbf{P}_{\mathrm{TA}} \left(\mathbf{u}(k-1) + \Delta\mathbf{u}(k)\right) + \Delta\mathbf{u}(k)^{\mathrm{T}} \mathbf{Q}_{\mathrm{TA}} \Delta\mathbf{u}(k) \right) \tag{3.18}$$

subject to

$$\tau_d(k) = \mathbf{B}\left(\mathbf{u}(k-1) + \Delta\mathbf{u}(k)\right) \tag{3.19}$$

$$\mathbf{u}_{\min} - \mathbf{u}(k-1) \leq \Delta\mathbf{u}(k) \leq \mathbf{u}_{\max} - \mathbf{u}(k-1) \tag{3.20}$$

$$\Delta\mathbf{u}_{\min} \leq \Delta\mathbf{u}(k) \leq \Delta\mathbf{u}_{\max} \tag{3.21}$$

where $\Delta\mathbf{u}(k) \in \mathbb{R}^{4\times1}$ is the change in force per actuator at time step k; $\mathbf{u}(k-1) \in \mathbb{R}^{4\times1}$ is the force per actuator at time step $k-1$; $\mathbf{P}_{\mathrm{TA}} \in \mathbb{R}^{4\times4}_{>0}$ and $\mathbf{Q}_{\mathrm{TA}} \in \mathbb{R}^{4\times4}_{>0}$ are positive definite penalty matrices; $\tau_d \in \mathbb{R}^{3\times1}$ is the vector with reference internal forces and moments, and is the output of the PI control and input of the thrust allocation; $\mathbf{B} \in \mathbb{R}^{3\times4}$ is the thrust allocation matrix; $\mathbf{u}_{\min} \in \mathbb{R}^{4\times1}$, $\mathbf{u}_{\max} \in \mathbb{R}^{4\times1}$, $\Delta\mathbf{u}_{\min} \in \mathbb{R}^{4\times1}$ and $\Delta\mathbf{u}_{\max} \in \mathbb{R}^{4\times1}$ are the limits of the actuators. The output of the thrust allocation is the control input at time step k: $\mathbf{u}(k) = \Delta\mathbf{u}(k) + \mathbf{u}(k-1)$. The values of the parameters used can be found in Chapter 6. This method is a simplification of the model predictive thrust allocation algorithm that has been designed in (Haseltalab et al. (2019)).

3.5.2. Option 2: FL MPC

As discussed in Chapter 2, Model Predictive Control (MPC) is a widely used control approach. Using a dynamic model of the system, the control actions over a finite time-horizon are being optimized, but only the action of the current timeslot is implemented. Keeping track of the future timeslots ensures that future events can be anticipated. It is essential to use a correct dynamic model, since all the actions are based upon this model. In the specific case of Roboat, a model is available, as discussed in Section 3.2. Therefore MPC is regarded as suitable for this project.

A downside of MPC is the high computation time. This makes the application of MPC in practice difficult. In literature this is mainly solved in two ways:

1. Use a linearized dynamic model (Chen et al. (2019a); Haseltalab et al. (2019))
2. Use a specialized MPC toolkit for code generation to make a very efficient code (Wang et al. (2018))

The first MPC method that is used in this research is based on Feedback Linearization. The model that is used, is the same as (3.1) -(3.2) and (3.8)-(3.9) except that disturbances are not taken into account, since no adequate solution to determine the disturbances is available in the test set-up. For the sake of Feedback Linearization, an auxiliary control input is introduced to establish a linear relationship between the system outputs and auxiliary inputs. Furthermore, the constraints of the optimization problem are linearized, which give the possibility to solve the problem using quadratic programming methods leading to significantly lower computational costs compared to the non linear optimization problem. The FL MPC control method is based on the velocity dynamics that can be rewritten in the following system, where the dependency on time is not shown for readability reasons:

$$\dot{\nu} = \mathbf{M}^{-1}\Big(\tau - \mathbf{C}(\nu)\nu - \mathbf{D}\nu\Big) \tag{3.22}$$

The following FL control law linearizes the system:

$$\tau = \mathbf{M}\Big(A\nu + B\zeta\Big) + \mathbf{C}(\nu)\nu + \mathbf{D}\nu \tag{3.23}$$

where $\zeta \in \mathbb{R}^{3\times 1}$ is the input vector of the linear closed-loop system, ν is the velocity vector and represents the states of the linear closed-loop system, and $A \in \mathbb{R}^{3\times 3}$ and $B \in \mathbb{R}^{3\times 3}$ are states and input matrices of the linear system, respectively. The values of these two matrices can be chosen to the desire of the designer as long as it leads to stability of the created linear closed-loop system[1]. The velocity dynamics can now be rewritten to the following linear closed-loop system:

$$\dot{\nu} = A\nu + B\zeta \tag{3.24}$$

This system can be discretized in the same way as is done in (3.8)-(3.9), the new matrices of the state-space representation are now indicated with A_d, B_d. Since this system contains only velocities and no positions, the dynamic system is smaller and thus the computation time is lower. However to accomplish this, the reference position must be transformed to a reference velocity. To this purpose, the reference velocity is approached in two steps. First the average velocity that is required between two time steps is determined. Then this is extrapolated to the required velocity at the second time step, based on the velocity at the first time step and under the assumption that the acceleration is constant:

$$\overline{\nu}_d(k+i|k) = \mathbf{R}^{-1}(\psi(k))\frac{\eta_d(k+i) - \hat{\eta}(k)}{\beta i t_s} \qquad i = 1,\dots,N_p; \tag{3.25}$$

$$\nu_d(k+i|k) = 2\Big(\overline{\nu}_d(k+i|k) - \hat{\nu}(k)\Big) + \hat{\nu}(k) \qquad i = 1,\dots,N_p; \tag{3.26}$$

where $\overline{\nu}_d(k+i|k) \in \mathbb{R}^{3\times 1}$ is the average reference velocity of time step $k+i$, calculated at time step k; $\mathbf{R}(\psi(k)) \in \mathbb{R}^{3\times 3}$ is the rotation matrix; $\eta_d(k+i) \in \mathbb{R}^{3\times 1}$ is the reference position at time step $k+i$; $\hat{\eta}_d(k) \in \mathbb{R}^{3\times 1}$ is the measured position at time step k; $\beta \in \mathbb{R}_{>0}$ is a tuning parameter; $t_s \in \mathbb{R}_{>0}$ is the sampling time; $\nu_d(k+i|k) \in \mathbb{R}^{3\times 1}$ is the reference velocity of time step $k+i$, calculated at time step k; and $\hat{\nu}(k) \in \mathbb{R}^{3\times 1}$ is the computed velocity at time step k. The following trivial optimization system is now created:

$$J = \min_{\zeta} \sum_{i=1}^{N_p} \left(\Big(\nu(k+i) - \nu_d(k+i)\Big)^{\mathrm{T}} \mathbf{Q}_{\mathrm{FL}} \Big(\nu(k+i) - \nu_d(k+i)\Big)\right) \tag{3.27}$$

subject to

$$\nu(k+i+1) = A_d\nu(k+i) + B_d\zeta(k+i) \qquad i = 1,\dots,N_p; \tag{3.28}$$

$$\tau_{\min} \le \mathbf{M}\Big(A_d\nu(k+i) + B_d\zeta(k+i)\Big) + \mathbf{C}\big(\hat{\nu}(k)\big)\hat{\nu}(k) + \mathbf{D}\hat{\nu}(k) \le \tau_{\max} \qquad i = 1,\dots,N_p; \tag{3.29}$$

where $\mathbf{Q}_{\mathrm{FL}} \in \mathbb{R}^{3\times 3}_{>0}$ is a positive definite penalty matrix. The required forces and moments τ_d can now be approached using the following linearized equation that give the output of this controller:

$$\tau_d(k+i) = \mathbf{M}\Big(A_d\nu(k+i) + B_d\zeta(k+i)\Big) + \mathbf{C}\big(\hat{\nu}(k)\big)\hat{\nu}(k) + \mathbf{D}\hat{\nu}(k) \qquad i = 1,\dots,N_p \tag{3.30}$$

[1] During simulations and real experiments in this research, A was chosen to be zero and B was chosen to be the identity matrix.

Thrust allocation Just as was the case with the PI control method, the FLMPC control method calculates the required forces and moments acting on the center of gravity. The same thrust allocation method as used with the PI control method, will be used now. However since the control method is predictive, future required forces are available. These can be put in the thrust allocation algorithm to make it predictive as well. This is especially useful on vessels that have slow reacting actuators. The methods used is the following:

$$J = \min_{\Delta\mathbf{u}} \sum_{i=0}^{N_{\text{TA}}-1} \left[\left(\mathbf{u}(k-1) + \sum_{j=0}^{i} \Delta\mathbf{u}(k+j)\right)^{\text{T}} \mathbf{P}_{\text{TA}} \left(\mathbf{u}(k-1) + \sum_{j=0}^{i} \Delta\mathbf{u}(k+j)\right) + (N_{\text{TA}}-1)\Delta\mathbf{u}(k+i)^{\text{T}} \mathbf{Q}_{\text{TA}} \Delta\mathbf{u}(k+i) \right] \quad (3.31)$$

subject to

$$\tau_d(k+i) = \mathbf{B}\Big(\mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i)\Big) \qquad i = 1, \ldots, N_{\text{TA}}; \quad (3.32)$$

$$\mathbf{u}_{\min} \le \mathbf{u}(k+i-1) + \Delta\mathbf{u}(k+i) \le \mathbf{u}_{\max} \qquad i = 1, \ldots, N_{\text{TA}}; \quad (3.33)$$

$$\Delta\mathbf{u}_{\min} \le \Delta\mathbf{u}(k+i) \le \Delta\mathbf{u}_{\max} \qquad i = 1, \ldots, N_{\text{TA}}; \quad (3.34)$$

where $N_{\text{TA}} \in \mathbb{N}$ is the number of prediction steps for the thrust allocation. This method is a simplification of the model predictive thrust allocation algorithm that has been designed in (Haseltalab et al. (2019)).

3.5.3. Option 3: NMPC

In theory the best model predictive controller is nonlinear, since the dynamic model of the vessel is nonlinear. As mentioned earlier, the computation time of a nonlinear solver is high. A solution that is also mentioned before, is to use a specialized MPC toolkit for code generation to make a very efficient code. This is done for example in the research of (Wang et al. (2018)) and will be the idea of the third control method. Using the ACADO toolkit (Houska et al. (2011)), an NMPC control method is implemented. The objective function of the NMPC control method is the following:

$$J = \min_{\mathbf{u}} \sum_{i=1}^{N_p} \left(\Big(\eta(k+i) - \eta_d(k+i)\Big)^{\text{T}} \mathbf{Q}_{\text{NMPC}} \Big(\eta(k+i) - \eta_d(k+i)\Big) + \mathbf{u}(k+i)^{\text{T}} \mathbf{R}_{\text{NMPC}} \mathbf{u}(k+i) \right) \quad (3.35)$$

subject to

$$\mathbf{x}(k+i|k) = \hat{\mathbf{x}}(k) + \int_{kt_s}^{(k+i)t_s} f(\mathbf{x}, \tau)\,dt \qquad i = 1, \ldots, N_{\text{P}}; \quad (3.36)$$

$$\mathbf{x}(k+i) = \begin{bmatrix} \eta(k+i) \\ \nu(k+i) \end{bmatrix} \qquad i = 1, \ldots, N_{\text{P}}; \quad (3.37)$$

$$\tau_d(k+i) = \mathbf{B}\mathbf{u}(k+i) \qquad i = 1, \ldots, N_{\text{P}}; \quad (3.38)$$

$$\mathbf{u}_{\min} \le \mathbf{u}(k+i) \le \mathbf{u}_{\max} \qquad i = 1, \ldots, N_{\text{P}}. \quad (3.39)$$

where $\mathbf{Q}_{\text{NMPC}} \in \mathbb{R}^{3\times3}_{>0}$ and $\mathbf{R}_{\text{NMPC}} \in \mathbb{R}^{3\times3}_{>0}$ are positive definite penalty matrices; and $\hat{\mathbf{x}}(k) = \begin{bmatrix} \hat{\eta}(k)^{\text{T}} & \hat{\nu}(k)^{\text{T}} \end{bmatrix}^{\text{T}} \in \mathbb{R}^{6\times1}$ is the measured state vector. The other variables have been defined before. As can be noticed, disturbances are not taken into account in the NMPC control method as well, since they cannot be measured.

This control method does not require a separate thrust allocation algorithm, since this is already incorporated in the optimization problem. It is suggested by (Veksler et al. (2016)) that this will result in a more optimal control output. Since the dynamic model used in this control method resembles the real dynamic model of the Roboat more than the FL MPC control method, it is expected that this control method will result in more accurate results. On the other hand, the NMPC control method will probably result in a higher computation time, so in theory the FL MPC control method can make use of more frequent control loops.

3.6. Conclusion

Throughout this chapter, answers have been given on the first part of the research question *'How to design a flexible formation control strategy?'*. A control strategy for one fully autonomous vessel based on a modular control structure has been designed. The dynamics of Roboat have been described by the widely accepted model suggested by (Fossen (2011)), shown in Equations (3.1) - (3.2) whereof the parameters that have been determined by (Wang et al. (2018)) for the 1:4 scale Roboat have been adopted. In the designed control strategy, the navigation system measures the position of the Roboat and subsequently calculates the current velocity. The measured position and speed are then sent to the guidance and control systems. The guidance system determines a local trajectory, based on a global trajectory and the measured positions of the own vessel and objects such as other vessels or canal boundaries.

Three options for the control system have been provided. First a baseline PI control method is designed, together with a thrust allocation algorithm. The second option is a feedback linearization MPC algorithm, together with a predictive thrust allocation algorithm. Finally an NMPC algorithm has been designed that incorporates the thrust allocation problem.

In the next chapter, the designed control strategies for one vessel are expanded to formation control strategies.

4
Multiple Roboat control

In the previous chapter, control strategies for one vessel have been designed. These control strategies will be expanded to formation control strategies in this chapter. With that, the second and last part of the answer to research question *'How to design a flexible formation control strategy?'* will be provided.

First a look will be given into the desired configuration of formation. Then, the control architecture of the formation strategy will be discussed. Thereafter, focus will be on the communication between Roboats. These parts will come together in the last section of this chapter, where the design of formation control system is finalized. The decisions made in this chapter, will be partly explained by relevant parts of the literature review performed in Chapter 2.

4.1. Configuration of formation

In the performed literature review, described in Chapter 2, two main types of configuration have been described: the fixed and flexible configuration. In the introduction of this research (Chapter 1) it is explained that the three main advantages for formation control of Roboat units in Amsterdam are: 1) the autonomous vessels will give less nuisance to the other users of the canals, since they sail close to each other in formation; 2) the small autonomous vessels will be easier to observe for other vessels and they can be better monitored, leading to an increased level of safety; 3) a probably sooner authorized solution with one human-operated vessel as leader and autonomous vessels as followers is revealed.

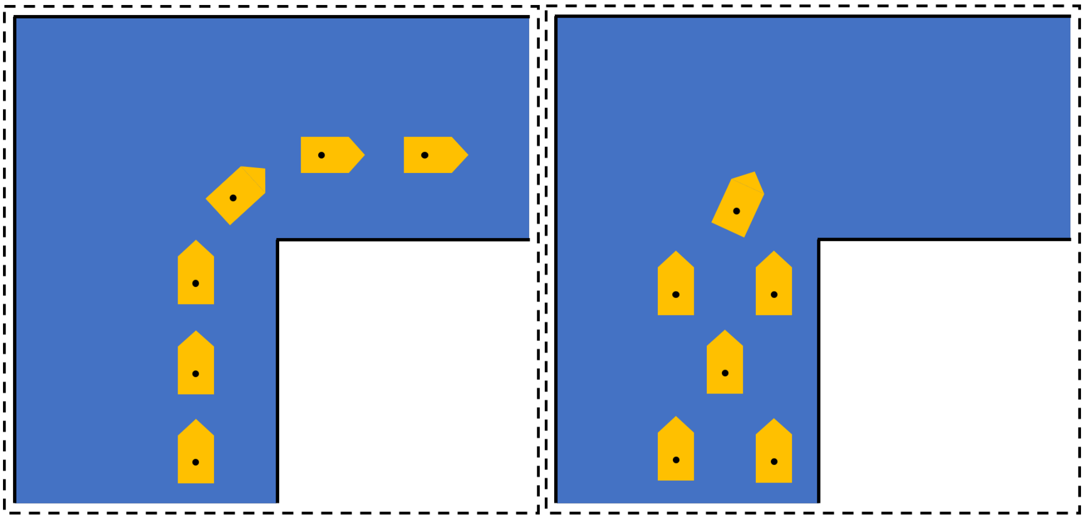
In light of the foregoing, a fixed configuration is not necessary to accomplish these foreseen advantages. Meanwhile the literature review states that the flexible configuration offers more options to deal with the (dynamic) surroundings. This is reason enough to choose for a flexible configuration of formation of Roboat units that will sail in the canals of Amsterdam. In this work the aspect of flexible configuration enables a Roboat unit to temporarily abandon the relative position constraint to avoid an object for instance.

In this thesis no specific research is done to the optimal position of Roboat units in the flexible configuration. This is regarded as an opportunity for follow-up research. It is argued that, to increase the benefits as much as possible, two types of flexible configurations are desired. The first type is a train-like configuration, such as the VTF described in (Chen et al. (2019b); Novimar (2017)). The second type is a diamond-like configuration. Both configurations are shown in Figure 4.1.

4.2. Control architecture

In the literature review (Chapter 2), three main approaches for formation control have been listed: Behavior-based; virtual structures; and leader-follower approaches. Other approaches that have been named in that chapter are graph theory, path guided, cluster space, and VTF. Literature does not suggest that one approach is better than the other. For the desired benefits of the formation control, listed in previous section, a leader-follower approach appears as the best fit. Mainly the third benefit - *A solution with a human-operated vessel as leader and autonomous vessels as followers will probably be authorized sooner* - works in favour of the leader-follower approach.

None of the works considered in the literature review describe a research with distributed leader-follower approach. From this it can be concluded that research to the leader-follower approach has been mainly



(a) Train-like formation (b) Diamond-like formation

Figure 4.1: Schematic overview of the two desirable flexible configurations for Roboat units in Amsterdam's canals

performed with centralized or decentralized approaches, while the literature review clearly stated that a distributed control approach is more advantageous than a centralized or decentralized control approach. The nature of the leader-follower approach is probably the reason for the lack of research on distributed leader-follower approaches, since one leader vessel determines (most of) the reference trajectory for follower vessels. However the major advantage of scalability is retained in the distributed leader-follower approach. Therefore a distributed leader-follower approach will be researched in this thesis. The exact architecture of the distributed control will be determined later on in this chapter. Possible considerations include the use of single or multi-layer control and a parallel or serial scheme.

4.3. Design of formation control

In the leader-follower approach, the trajectory of the leader is the input of the trajectory planner of the followers. This means that at least the position of the leader must be communicated to the followers. Since in Chapter 3 predictive control methods have been designed, it is also desired that the predicted positions of the leader are communicated. Furthermore, the followers have to make sure that they do not collide with each other. Depending on the configuration of formation, the global reference trajectories of followers could come within the safety distance of each other. Collisions can be avoided by taking the current and predicted positions of other followers into account when creating local reference trajectories. In summary, every vessel in the formation has to share its position and predicted positions with the other vessels.

Based on the control architecture and communicated parameters that have been determined in the previous sections, an overview of the communication outline of the formation is given in Figure 4.2. As shown in the figure, the design of the control system of the leader Roboat is changed in two ways with respect to the design of the control system of a single Roboat (Chapter 3):

1. The local trajectory planner of the Roboat takes the current and predicted positions of the other vessels and objects into account;

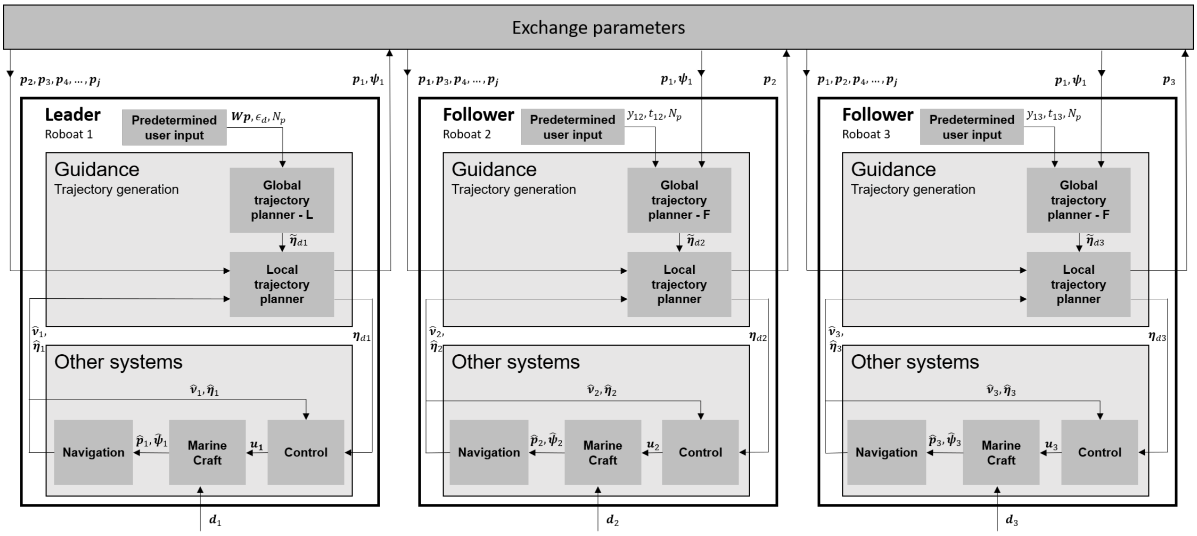
2. The Roboat sends its own current and predicted position to other Roboat units.

The design of the control system of the follower Roboat units is changed in two additional ways:

3. Waypoints and a desired speed are replaced by a desired headway and lateral distance between the leader and the follower Roboat as user input;

4. The current and predicted position of the leader Roboat is input of the global trajectory planner.

Items 1 and 2 do not have any impact on the design of the systems of the Roboat, only the communication has to be established. Items 3 and 4 do influence the design of the global trajectory planner. The design of the global trajectory planner for follower Roboat units will be discussed in the next paragraph.



Declaration of symbols		
$\boldsymbol{p}_j(i)$	$\in \mathbb{R}^{2x1}$	Predicted and current position of vessel (or object) j
y_{1j}	$\in \mathbb{R}$	Desired lateral distance between vessel 1 and j
t_{1j}	$\in \mathbb{R}$	Desired headway between vessel 1 and j
$\hat{\boldsymbol{\eta}}$	$\in \mathbb{R}^{3x1}$	Measured position
$\hat{\boldsymbol{\nu}}$	$\in \mathbb{R}^{3x1}$	Measured velocity
$\boldsymbol{Wp}$	$\in \mathbb{R}^{2xn_{Wp}}$	Waypoints ($n_{Wp} \in \mathbb{N}$; # of waypoints)
ϵ_d	$\in \mathbb{R}$	Reference surge speed
N_p	$\in \mathbb{N}$	Number of prediction steps
$\tilde{\boldsymbol{\eta}}_d(i)$	$\in \mathbb{R}^{3x1}$	Global reference trajectory
$\boldsymbol{\eta}_d(i)$	$\in \mathbb{R}^{3x1}$	Reference trajectory
$\boldsymbol{u}$	$\in \mathbb{R}^{4x1}$	Control input
$\boldsymbol{\tau}$	$\in \mathbb{R}^{3x1}$	Internal forces and moments
$\boldsymbol{d}_j$	$\in \mathbb{R}^{3x1}$	External forces and moments applied to vessel j
$\hat{\boldsymbol{p}}$	$\in \mathbb{R}^{2x1}$	Measured position w/o heading
$\hat{\boldsymbol{\psi}}$	$\in \mathbb{R}$	Measured heading
$\boldsymbol{\psi}_j(i)$	$\in \mathbb{R}$	Predicted and current heading of vessel j

Figure 4.2: Schematic overview of the communication between Roboat units in formation

Global trajectory planner - follower Roboat units

To create a global trajectory that follows the trajectory of the leader Roboat, the following calculation steps are performed:

1. A matrix containing the previous, current, and predicted positions of the leader per time step is made;
2. The first time step that will be taken into account is determined as the current time step minus the headway divided by the sampling time;
3. Per time step, the desired lateral distance between the leader Roboat and follower is transformed to the body-fixed frame and added to the reference trajectory.

This leads to the following equations where j indicates the number of the follower Roboat:

$$n_{1\mathrm{j}} \approx \frac{t_{1\mathrm{j}}}{t_s} \tag{4.1}$$

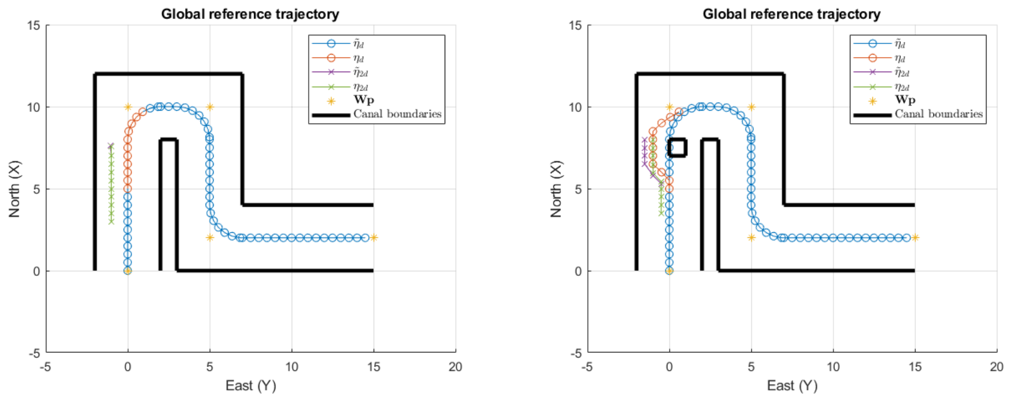
$$\eta_1 = \begin{pmatrix} \mathbf{p}_1 \\ \psi_1 \end{pmatrix} \tag{4.2}$$

$$\eta_1^b(a) = \mathbf{R}^{-1}\big(\psi_1(a)\big)\eta_1(a) \qquad i = 1,\ldots,N_p; \qquad a = k+i-n_{1\mathrm{j}}; \tag{4.3}$$

$$\tilde{\eta}_d(k+i) = \mathbf{R}\big(\psi_1(a)\big)\left(\eta_1^b(a) + \begin{pmatrix} 0 \\ y_{1\mathrm{j}} \\ 0 \end{pmatrix}\right) \qquad i = 1,\ldots,N_p; \qquad a = k+i-n_{1\mathrm{j}}. \tag{4.4}$$

where $t_{1\mathrm{j}} \in \mathbb{R}_{>0}$ is the desired headway in seconds; $n_{1\mathrm{j}} \in \mathbb{R}_{>0}$ is the desired headway in number of time steps; and $y_{1\mathrm{j}} \in \mathbb{R}$ is the desired lateral distance between the leader and the follower Roboat. The other variables have been defined before.

It is noted that without local trajectory planner, or with an adapted local trajectory planner, a fixed configuration of formation can be created. However if a fixed configuration is desired, a feedback to the leader from the follower should be implemented, to ensure that the followers can keep up with the leader's speed and that the created trajectory is navigable for the followers as well. The operation of the global and local trajectory planners of a follower Roboat are shown in Figure 4.3. The operation of the global and local trajectory planner of the leader Roboat is shown in Chapter 3, where a similar figure is shown: Figure 3.8. Figure 4.3a shows that



(a) Without extra object

(b) With an extra object added

Figure 4.3: Graphically displayed output of the local trajectory planner of two Roboat units at $k = 11$ (or $t = 5$[s]), with inputs $v_{\text{max}} = 1.5$[m/s], $d_{\text{safe}} = 1$[m], $N_p = 10$[−], $y_{1j} = -0.5$[m], $t_{1j} = 1$[s] and the global reference trajectory of the example in Figure 3.7

the local and global trajectory of the follower is equal and on the expected location with these input parameters. Figure 4.3b shows that the global trajectory planner has made a trajectory that has the desired offset from the leader's trajectory. However this produced global reference trajectory comes within the safety distance of a canal boundary. Therefore the local trajectory planner has changed the reference trajectory slightly.

Furthermore, it should be clear that with the designed global trajectory planner, both configurations that have been discussed in Section 4.2 and are shown in Figure 4.1 can be achieved.

4.4. Conclusion

This chapter has complemented the previous chapter in answering research question *'How to design a flexible formation control strategy?'*. During the chapter two configurations of formation have been classified as desired: the flexible train-like configuration and the flexible diamond-like configuration. To reach the goal such as defined in Chapter 1[1], a leader-follower approach is suggested and in line with literature recommendations, a distributed control architecture is adopted. In order to perform leader-follower formation control and at the same time avoid collisions, all vessels in the formation exchange their current and predicted positions.

The control strategies that have been designed in the previous chapter can be used for the leader Roboat with only slightly modifications to establish the communication. For the follower Roboat units, the design of the global trajectory planner, located in the guidance system, changes to be able to follow the leader Roboat.

In the next chapter, focus will be put on the implementation of the formation control strategies on 1:4 scale Roboat units. This includes the technical details of the Roboat, the command structure and software platform.

[1]The goal of this research, defined in Chapter 1 is to design a strategy that is able to let multiple autonomous Roboat units sail in a flexible formation in the canals of Amsterdam for the application of transporting household waste

5

Implementation of the formation control strategy

In the previous chapters, a formation control strategy has been designed. The intention of this research is to bring the formation control strategy into practice. For this purpose, this chapter deals with research question *'How to implement the formation control strategy in Roboat units?'*. This question will be answered by first stating the implementation challenges that resulted of the literature review, so that these can be taken into account in the design. Secondly, the technical details of Roboat will be listed, as these also constrain the design. Finally, the design of the command architecture is discussed.

5.1. Implementation challenges

In literature, multiple challenges that have been holding back real experiments have been identified, including:

1. **External disturbances & parameter perturbation**: In (Li et al. (2020)) it is pointed out that the influence of external disturbances and parameter perturbation can be crucial if not considered in the control algorithms;

2. **Parameter sensitivity**: In (Mas and Kitts (2014)) the importance of a correct model is discussed. In their research, the formation started showing signs of instability when the model parameters deviated more than 140% from the ideal parameters.;

3. **Computation time**: A high computation time is often mentioned as implementation challenge for MPC methods (Li et al. (2020)). However none of the research works that have been validated with real experiments - listed in Table 2.4 - give information on the frequency of the control loop.

While challenges have been defined in literature, no sufficient practical information on performed real experiments has been released. Especially information on the frequency of control loops would have been very helpful. Since this information lacks in literature, it will be found out in this research.

5.2. Technical details of Roboat

The real experiments are carried out using 1:4 scale Roboat units. In this section, the technical details of this vessel will be shared, in order to make this research replicable. The characteristics and properties of the Roboat units that are possessed by AMS Institute have been listed in Table 5.1.

The simplified schematic overview of the components and their connection in the 1:4 scale Roboat, is shown in Figure 5.1. Non-essential items such as a voltage checker have been omitted. The RTK GPS, IMU, Wi-Fi antenna, and data processor are connected to the mini pc via USB cables. These units are also powered via the USB connection. The mini pc itself requires 19V and the thrusters require 12V. These voltages are supplied by the power converter, that ensures equal output voltages if its input voltage is above 9V.

Table 5.1: Properties of the 1:4 scale Roboat

Sector	Property	Value / type
Physical properties	Dimensions	0.90 by 0.45 by 0.15 [m]
	Total mass	9.2 [kg]
Sensors	RTK GPS	Emlid Reach RS+
	IMU	MicroStrain 3DM-GX5-25
Computer & communication	Wi-Fi antenna	Alfa Network AWUS036NH
	Mini pc	Intel Dawson Canyon NUC7i7DNH
	Data processor	STM32F103
Actuators & power	Thrusters	4 x T100 Thruster
	Battery	11.1 V rechargeable Li-Po
	Power converter	9-36V to 12V & 19V
	Operation time	2 [h]

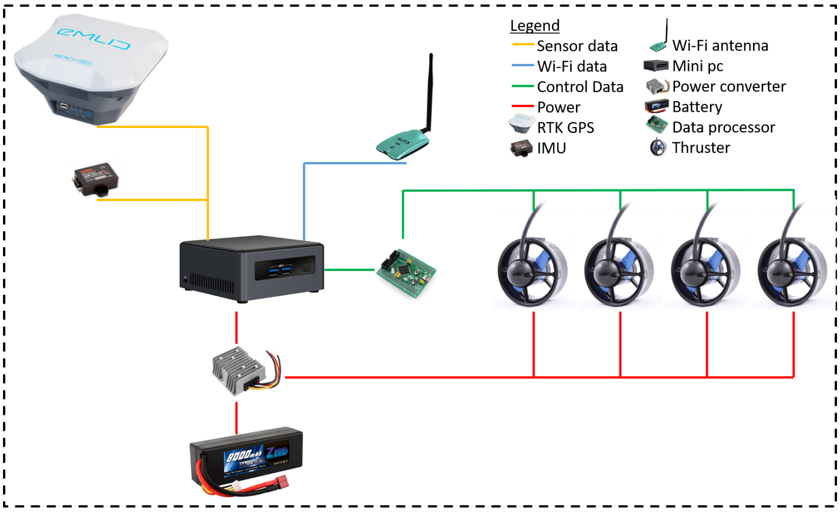


Figure 5.1: Simplified overview of the components and their connection in the 1:4 scale Roboat

5.3. Implemented command architecture

The algorithms that have been designed in Chapters 3 and 4 have to be implemented in real Roboat units. This can be done in various ways and this certainly influences the performance of the formation control strategy. In this section, first the way of communication will be covered, than the programming languages used will be examined and finally a scheme of the complete implementation will be given.

5.3.1. Communication between Roboat units

The Roboat units have to be able to communicate with each other as has been discussed in Chapter 4. Moreover, during the real experiments it is also desired to be able to take over control from the quay using a joystick. A system that is able to let a practical unlimited amount of robots communicate with each other is ROS, short for Robotic Operating System (Stanford Artificial IntelligenceLaboratory (2018)). Previous experiments that have been performed using Roboat units, have also used ROS. Although that software cannot be used in this thesis - so all software has to be self-written - it is a logical step to also use the ROS framework.

5.3.2. Programming languages

An environment that is compatible with ROS, straightforward to use and that allows a convenient way of analyzing results is Matlab. Many of the discussed research works in the literature overview have done simulations in Matlab. However, it is also noted that Matlab is slow in comparison to solutions written in C or C++. Besides, the ROS environment is compatible with Matlab, but works best with code written in C++. At this point the choice has been made to write all the code on the on-board computers in C++, since that works best with ROS. Since the developed algorithms are experimental, it is chosen to write that code in Matlab, to

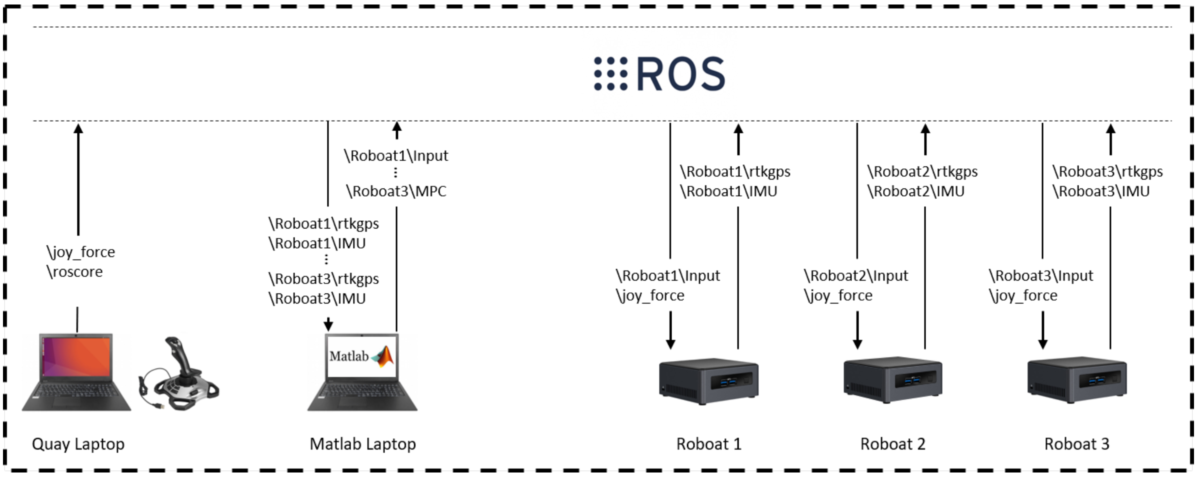


Figure 5.2: Global overview of the communication of the Quay laptop, Matlab laptop and Roboat units via ROS topics

be able to optimize and adjust the code easily. When the formation control strategy works, the MATLAB code can be rewritten to C++ code to lower the computation time.

5.3.3. Command structure

A big consequence of using Matlab, is that a central computer with Matlab on it, is required. That central computer determines the control actions of the Roboat units. This implies that the distributed control approach is simulated in the real experiments. In other words, the code on the central computer is written in such a way that the approach is still distributed control.

The ROS structure is globally shown in Figure 5.2. In the figure, the ROS topics that are being published by the various components are shown. It is shown that the *Quay laptop* sets up the ROS network. Furthermore a joystick is attached to the *Quay laptop* and is able to take over control in emergencies. The Roboat units determine their current positions and send those to the ROS network. Subsequently the *Matlab laptop* picks up those positions, executes the complete control loop, including the determination of predicted positions, and sends the control actions to the Roboat units. An extension of Figure 4.2 - which is placed in the previous chapter - is given in Figure 5.3. This figure provides a very detailed overview of the command structure and shows the division between Matlab and C++ code. As can be seen in the figure, most parts of the complete system are covered in the Matlab code. Only the code that is on the Roboat itself - the marine craft -, and the code for the joystick on the *Quay laptop* are written in C++.

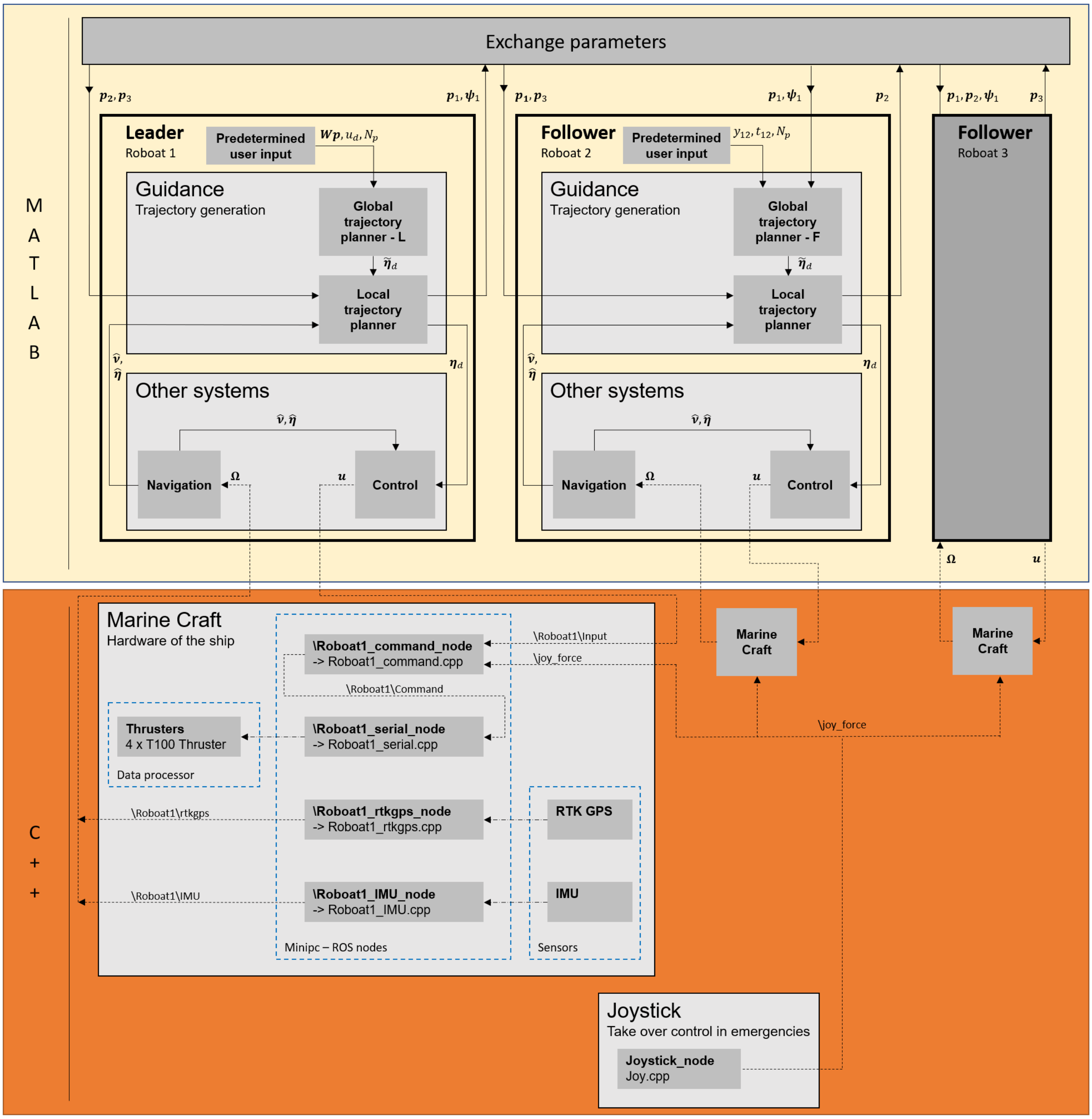
The Matlab control loop is executed at 2 Hz. This is sufficient to control the Roboat units at low speeds. However, when higher speeds are desired, the control loop rate should be higher. The information rate of the IMU sensor is 100 Hz, and the RTK GPS unit provides its information at a rate of 5 Hz. So the Matlab control loop is the bottleneck now. This can be improved by coding this loop in C++ and implement this code on the mini pc on Roboat itself. In this way, the formation control approach will be really distributed rather than simulated distributed.

5.4. Technical details of the laptops used

As shown in the previous part of this chapter, two laptops are required for performing real experiments with the Roboat units: the Quay laptop and the Matlab laptop. During the simulations, the Matlab laptop simulates the behaviour of the Roboat units, so only the Matlab laptop is required then. The characteristics of both laptops have impact on the computation time of the control loop, therefore the technical details of both laptops are listed in this section.

The operating system of the Matlab laptop is Windows 10 and Matlab version R2019b is used. The laptop uses a quad core 1.8 GHz Intel(R) Core(TM) i7-8550U processor and has 8 GB of RAM.

The operating system of the Quay laptop is Ubuntu 18.04.3 LTS. The main program that is run on this laptop is ROS Melodic Morenia. The laptop makes use of the same type of processor as the Matlab laptop: a quad core 1.8 GHz Intel(R) Core(TM) i7-8550U processor. Besides, the laptop has 16 GB of RAM.



Declaration of symbols			$\hat{\eta}$	$\in \mathbb{R}^{3x1}$	Measured position	N_p	$\in \mathbb{N}$	Number of prediction steps	τ	$\in \mathbb{R}^{3x1}$	Internal forces and moments	**Legend**	
p_j	$\in \mathbb{R}^{2x(N_p+1)}$	Predicted and current position of vessel (or object) j	$\hat{\nu}$	$\in \mathbb{R}^{3x1}$	Measured velocity	$\tilde{\eta}_d$	$\in \mathbb{R}^{3x(N_p+1)}$	Global reference trajectory	d_j	$\in \mathbb{R}^{3x1}$	External forces and moments applied to vessel j		Serial communication
y_{1j}	$\in \mathbb{R}$	Desired lateral distance between vessel 1 and j	Wp	$\in \mathbb{R}^{2xn_{Wp}}$	Waypoints ($n_{Wp} \in \mathbb{N}$; # of waypoints)	η_d	$\in \mathbb{R}^{3x(N_p+1)}$	Reference trajectory	Ω		Sensor data		Rostopic
t_{1j}	$\in \mathbb{R}$	Desired headway between vessel 1 and j	u_d	$\in \mathbb{R}$	Reference surge speed	u	$\in \mathbb{R}^{4x1}$	Control input	ψ_1	$\in \mathbb{R}^{1x(N_p+1)}$	Predicted and current heading of vessel 1		Matlab variables

Figure 5.3: Detailed overview of the various parts of code and the division of code in Matlab and C++

5.5. Conclusion

The algorithms designed in Chapter 3 and 4 have been implemented in Roboat using the ROS framework. The distributed control approach is simulated in the Matlab code that runs on a central laptop at the quay. The codes that have been developed on the mini computers on-board Roboat are written in C++. Furthermore a *Quay laptop* that runs the ROS network is used to attach a joystick that can take over control in emergencies.

The next step is to experimentally assess the described formation control strategies. This will be done in the next part of this research. First, in Chapter 6, the methods that have been used to experimentally assess the strategies will be handled.

6

Methods

In the previous chapters, formation control strategies have been designed and implemented in 1:4 scale Roboat units. It is time to evaluate their performance, which is done in this part of the research. This chapter contains the explanation of the evaluation methods, while Chapter 7 contains the results and analyses. Together, the two chapters provide answers to the research question *'What is the performance of the designed formation control strategy and its implementation in Amsterdam's canal network?'*.

In this chapter, first focus is put on the question how to evaluate the performance of the designed formation control strategies. Afterwards, the simulation set-up is discussed, including the layout of the test environment and the parameters used for the control algorithms. Then, the set-up for the real experiments is handled, which includes an overview of the parameters used and the accuracy of the sensors of the Roboat units.

6.1. Performance evaluation

In this section, literature on maritime formation control will be consulted to learn from the methods used to evaluate the performance of their strategies. Based on that, the approach to evaluate the performance of the designed strategies is discussed afterwards.

6.1.1. Frequently used KPIs in formation control

The Key Performance Indicators (KPIs) that have been used in former research works on maritime formation control validated with real experiments, are listed in Table 6.1. KPIs that have been used in research works on maritime formation control validated with simulations, can be found in Table 6.2. Most research works use graphs to show the position, speed and control inputs. It is worth to note that only in (Kyrkjebø et al. (2007); Li et al. (2020)) results are presented both in quantitative and graph form.

6.1.2. The performance of formation control strategies in literature

Since most research works only show graphs that display their KPIs, it is hard to express the results quantitatively. However it seems very valuable to compare the results of this research with results from other research works. Therefore an indication of the trajectory tracking performance of the research works shown in Table 6.1 will be given in the next paragraphs. It should be kept in mind that the experimental conditions, layouts, and assigned tasks per experiment differ greatly.

Real experiments The trajectory tracking error of the experiments of (Li et al. (2020)) fluctuates frequently from -5 to 5 m, with mean values shown in Table 6.1. The trajectory error of the center of formation of the experiments mentioned in (Mas and Kitts (2014)) fluctuates frequently as well from -5 to 5 m. However the error of the desired distance between the vessel fluctuates more heavily from -7.5 to 7.5 m. The position error of the follower in the straight line experiment of (Schoerling et al. (2010)) fluctuated from -0.6 to 0.3 m, in the circular motion test it fluctuated from -1 to 3 m and in the zigzag test from - 0.2 to 0.8 m.

In (Arrichiello et al. (2010)) a leader-follower formation strategy is only activated when the distance between the two vessels is near the maximum communication range. The KPIs are impossible to estimate since no reference trajectory is shown. The graphs shown in (Mahacek et al. (2009)) are very hard to read, it appears

Table 6.1: KPIs used in maritime formation control research works validated with real experiments

References	Formation shape	Trajectory	Desired speed	KPI	Performance
Li et al. (2020)	Triangular	Straight line	7.7 [m/s]	Transverse error Longitudinal error Heading angle error Velocity error	-3 to 3 [m] -4 to 4 [m] -2 to 2 [m] -0.3 to 0.3 [m/s]
Mas and Kitts (2014)	Various	Waypoints	0.3 [m/s]	Cluster space parameters	Various (in graphs)
Schoerling et al. (2010)	Triangular	Straight line	0.5 [m/s]	Control input, Position & Speed	Various (in graphs)
		Circular motion	0.5 [m/s]	Control input, Position & Speed	Various (in graphs)
		Zigzag motion	0.5 [m/s]	Control input, Position & Speed	Various (in graphs)
Arrichiello et al. (2010)	None	Waypoints	0-1 m/s	Position, Relative distance & Speed	Various (in graphs)
Mahacek et al. (2009)	Triangular	Straight line	0.4 m/s	Position	Various (in graphs)
Kyrkjebø et al. (2007)	Parallel	Sinusoidal	0.2 m/s	Position & Estimated position Mean error X position Maximum error X position	Various (in graphs) 0.079 [m] 0.105 [m]

that the trajectory tracking error is between - 10 and 10 m. In (Kyrkjebø et al. (2007)) the position error of the follower is in the initial phase about 2.2 m and converges to about 0.1 m. The error does not fluctuate a lot.

Simulations The formation tracking errors of the followers in the simulation of (Peng et al. (2019)) are less than 0.05 m during normal operation and fluctuate slightly. Due to object avoidance the formation tracking errors become up to 2 m. The formation tracking error in the simulation of (Riahifard et al. (2019)) converge to 1 m after the initial phase. The path following errors in the simulation of (Chen et al. (2019c)) are up to 15 m. However many vessels take part in this simulation and perform collision avoidance. The simulation performed in (Abdelaal et al. (2016)) mainly concerns collision avoidance and the results are non comparable with other studies.

Conclusion Results from both real experiments and simulation differ a lot. This can be partly explained by the different circumstances of the tests and properties of the vessels used. The performance of the other research works give however a very rough range in which the results of this research should be. From the real experiments it appeared that the range varies from a constant error of 0.1 m to a fluctuating error between -10 and 10 m. In simulation the range is from a slightly fluctuating error of less than 0.05 and a constant error of 1 m.

6.1.3. Approach for experimental assessment

For the application of the formation control strategy in Amsterdam's canals, the strategy needs to be assessed on several aspects. First of all, the strategy should enable the tracking of a trajectory with turns both to the right and left, and an object should be avoided during the trajectory. In the canals of Amsterdam the maximum speed of the full scale Roboat is 6 km/h, or 1.7 m/s. Using Froude scaling the corresponding speed of the 1:4 scale Roboat is equal to $1.7/\sqrt{4} = 0.85$ m/s. However the designs of the full scale and 1:4 scale Roboat differ fairly in the favour of the full scale. In (Wang et al. (2018)) a maximum speed of 0.69 m/s for the 1:4 scale Roboat is assumed and in their experiments a desired speed of 0.20 m/s is used.

Based on the KPIs used in literature (see Tables 6.1 and 6.2) the following graphs will be used to evaluate the performance:

1. Position over time including reference position

Table 6.2: KPIs used in maritime formation control research works validated with simulations

References	Formation shape	Trajectory	Desired speed	KPI	Performance
Peng et al. (2019)	Circles around leader	Straight line	0.1-0.6 [m/s]	Position, Position errors, Speed, Estimation of speed & Distance to obstacles	Various (in graphs)
Riahifard et al. (2019)	Triangular	Straight line	0.4 [m/s]	Position, Position error, Control input	Various (in graphs)
Chen et al. (2019c)	VTF	Trajectory	3-4 [m/s]	Position, Position error, Speed & Total travel time	Various (in graphs)
Abdelaal et al. (2016)	None	Straight line	5.5 [m/s]	Position & Speed	Various (in graphs)

2. Speed over time
3. Control input over time
4. Distance to obstacles and other vessels over time

Furthermore it seems better to also quantitatively assess the control strategy. Besides, in this research, distinction between the performance of the guidance and control system is desired. Therefore, also the following KPIs will be taken into account per Roboat:

1. Root mean square error (RMSE) of the reference trajectory relative to the global trajectory (guidance system):

$$\sqrt{\frac{\sum_{k=1}^{N}\left(\left(\|\mathbf{p}_d(k)-\tilde{\mathbf{p}}(k)\|\right)^2\right)}{N}} \tag{6.1}$$

2. RMSE of the measured position relative to the reference trajectory (control system):

$$\sqrt{\frac{\sum_{k=1}^{N}\left(\left(\|\hat{\mathbf{p}}(k)-\mathbf{p}_d(k)\|\right)^2\right)}{N}} \tag{6.2}$$

3. RMSE of the measured heading relative to reference heading (control system):

$$\sqrt{\frac{\sum_{k=1}^{N}\left(\left(\|\hat{\psi}(k)-\psi_d(k)\|\right)^2\right)}{N}} \tag{6.3}$$

4. Square root of the total sum of forces applied by each thruster squared (control system):

$$\sqrt{\sum_{k=1}^{N}\left(\left(\|\mathbf{u}(k)\|\right)^2\right)} \tag{6.4}$$

5. Square root of the total sum of the differences in successive control inputs of each thruster over time squared (control system):

$$\sqrt{\sum_{k=1}^{N-1}\left(\left(\|\mathbf{u}(k+1)-\mathbf{u}(k)\|\right)^2\right)} \tag{6.5}$$

KPIs 1-3 are most important since they indicate how good the guidance system determines a trajectory and subsequently how good the control system follows that trajectory. KPIs 4-5 indicate the load on the actuators. Although those KPIs are not directly related with the power consumption, they are indirectly related. All KPIs can only be positive and the optimal value for all is zero.

In existing literature, a difference in behaviour between the steady state and start-up phase often exists. Therefore these KPIs will be calculated over a part of, and over the complete duration of the experiment.

Figure 6.1: The dedicated autonomous vessel testing area that may be used by AMS Institute in Amsterdam

6.2. Simulation set-up

The first method of assessment of the designed strategies is to simulate the strategy and evaluate the performance in simulation using the aforementioned graphs and KPIs. In this section first the layout of the test environment will be discussed. The section concludes with an overview of the parameters used.

6.2.1. Layout of test environment

To be able to compare simulation and real experiment results, the same set-up for both is desired. Therefore the part of Amsterdam's canal network that is dedicated as testing area for AMS Institute is used for the simulations. This area is indicated on the map of Amsterdam shown in Figure 6.1. To test the ability of making right and left turns, the behaviour during a straight line trajectory, and object avoidance application, a desired trajectory that looks as an 'eight' is proposed. This desired trajectory with corresponding waypoints is shown in Figure 6.2. Virtual objects can be added in the layout. In simulation tests will be performed using three Roboat units: one leader and two followers.

6.2.2. Simulated system

The system that has been described by Equations (3.8) - (3.9) is simulated in MATLAB using the ode45 solver to simulate the behaviour of a Roboat. The corresponding model parameters that are used, have been discussed in Section 3.2 and are shown in Table 3.1. No disturbance or random noise is added to the simulated system in order to be able to compare the control strategies with equal conditions. The technical details of the laptop used, can be found in Section 5.4.

6.2.3. Values of parameters used

In simulation, six tests are performed in which the three different control systems on all Roboat units as well as two variations of the guidance system of the leader Roboat are tested. In the first three tests, the guidance system of the leader is the in Section 3.4 presented guidance system consisting of a global and local trajectory planner. This guidance system is also used for the followers in all tests. In the second three tests, only the global trajectory planner of the leader is used. These six tests have been chosen to assess the performance of the three control systems and the guidance system. An overview of the tests is provided in Table 6.3. The parameters that are used in the six tests, are shown in Table 6.8. The tuning parameters of the PI, FL MPC, and NMPC control systems are shown in Tables 6.4, 6.5 and 6.6 respectively.

The tuning parameters of the PI control algorithm have been determined according to the method described in (Haugen (2012)). The parameters of the FL MPC and NMPC algorithm have been determined experimentally.

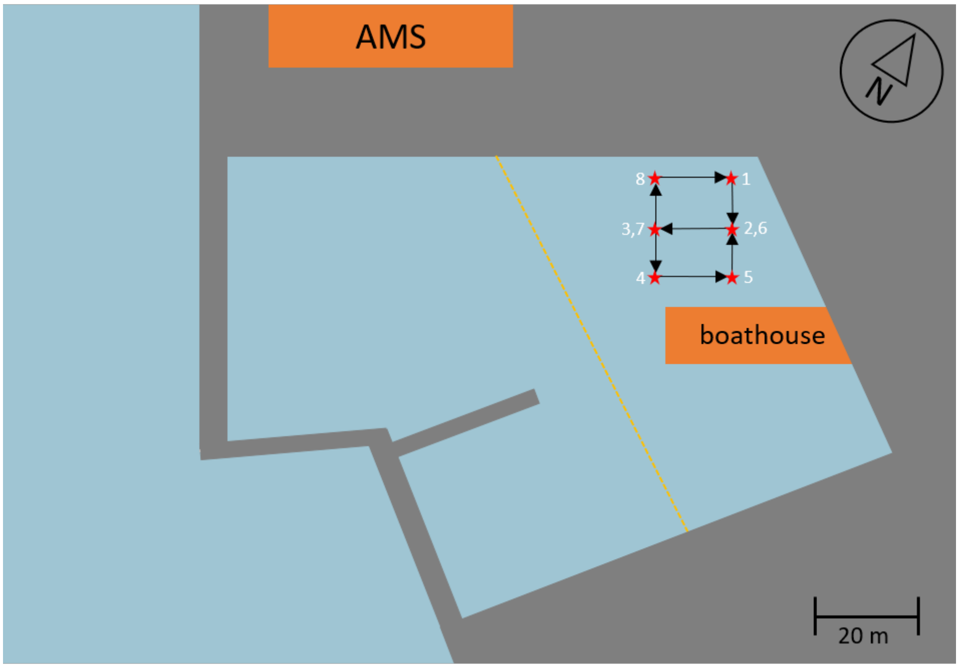


Figure 6.2: Desired trajectory of the leader of the formation, including the corresponding waypoints, canal boundaries, and location of the building of AMS Institute

6.3. Experimental set-up

After the simulation tests, the designed strategies have been assessed in real experiments. In this section, the set-up of these experiments is discussed. As mentioned in the previous section, the layout of the test environment in the real experiments is the same as that of the simulations. Details of the layout are discussed in Section 6.2.1. In this section the parameters that have been used for the real experiments will be given. Furthermore, a test to determine the accuracy of the sensors of the Roboat units is performed and will be handled in this section.

6.3.1. Values of parameters used

Five real experiments have been performed to evaluate the formation control strategies. Compared to the simulation tests, the test with PID control system and complete guidance system is not performed, since the guidance system appeared to be incompatible with the PID control strategy in practice. This is discussed in the next paragraph. An overview of the performed experiments can be found in Table 6.7. The parameters that have been described in Table 6.8 are also used in the real experiments. The tuning parameters that have been used in the experiments, can be found in Tables 6.4, 6.5, and 6.6. In the following paragraphs the method of determining the tuning parameters of the three control algorithms is described.

PI control In order to tune the PI control system parameters, a test is done with one Roboat. The guidance system of the Roboat consists only of the global trajectory planner and the input is the same as during the other tests. Using the measured position over time, the overshoot in x, y, and ψ direction is determined. Using the overshoot, the tuning parameters have been determined according to the method described in (Haugen (2012)). The final parameters are shown in Table 6.4 and the results of the test with these parameters have been included in Appendix D. In total 7 tests have been carried out to tune the PI parameters.

Afterwards, the PI control system has been tested with the full guidance system. In these tests it appeared that the Roboat was not able to track the trajectory. A plausible explanation is that the parameters were set too low to provoke a control input that could move the Roboat over the maximum distance that is limited by the guidance system. The error that is the input for the control system - the difference between the reference trajectory and actual position - is limited by the guidance system, since the next waypoint has to be reachable when sailing at maximum speed. However a correct setting of the PI parameters could not be found. The overshoot became too large when the parameters were set higher. After several tests it is concluded that the

Table 6.3: An overview of the six simulation tests

Category	Parameter	Value					
		Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Control system settings	Control method	PI	FL MPC	NMPC	PI	FL MPC	NMPC
	Prediction horizon (N_p)	-	20 [-]	10 [-]	-	20 [-]	10 [-]
Guidance system settings	Guidance system leader	complete	complete	complete	global trajectory only	global trajectory only	global trajectory only
	Guidance system followers	complete	complete	complete	complete	complete	complete

Table 6.4: Tuning parameters for PI control

Parameter	Simulation	Real experiments
K_p	$\begin{bmatrix} 3.5 & 0 & 0 \\ 0 & 3.0 & 0 \\ 0 & 0 & 2.0 \end{bmatrix}$	$\begin{bmatrix} 0.96 & 0 & 0 \\ 0 & 0.96 & 0 \\ 0 & 0 & 0.56 \end{bmatrix}$
K_i	$\begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{bmatrix}$	$\begin{bmatrix} 0.017 & 0 & 0 \\ 0 & 0.017 & 0 \\ 0 & 0 & 0.017 \end{bmatrix}$
K_d	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

PI control system is not compatible with the full guidance system.

FL MPC The FL MPC control system has two tuning parameters: $\mathbf{Q}_{FL}$ and β. These parameters are tuned while performing the same set-up as described for the determination of the tuning parameters of the PI control system: one Roboat sails a trajectory using only the global trajectory planner part of the guidance system. The initial parameters that have been tested are those used in simulation. After circa five tests the sailed trajectory was found to be good enough. The results of this test (with parameter $\beta = 10$) have been included in Appendix D.

After these first tests, around fifteen tests with the complete guidance system have been carried out. This led to a change of the parameter β. The final parameters can be found in Table 6.5.

NMPC The two tuning parameters of the NMPC control system - Q_{NMPC} and R_{NMPC} - have been determined in a same way as was done for the parameters of FL MPC control system. The initial parameters gave good results and a change of these did not result in an overall improvement of the control system. After circa five tests, it was decided to keep the initial parameters. These parameters can be found in Table 6.6. The results of the test with those parameters is included in Appendix D.

6.3.2. Accuracy test sensors

To assess the accuracy of the RTK GPS sensors and the IMUs of the Roboat units, two sets of experiments have been carried out. The first set included ten experiments. In each experiment, one Roboat is carried for a distance of 30 m, while maintaining a constant speed and heading. The results of the experiment can be found in Appendix E. In the experiments, the x coordinate of the Roboat was kept constant. While the mean absolute deviation around the mean of the measured x coordinate was between 0.03 and 0.12 m, the mean absolute deviation around the true value was between 0.45 and 3.83 m. The high deviations mostly occurred in the first four tests. However in tests 5-10 still high constant errors exist: 0.45 to 0.71 m. The main takeaway from these experiments is that the variable error of the position is small, while a constant error exists and changes over time (but on such a large time scale that is does not influence the variable error). The results are

Table 6.5: Tuning parameters for FL MPC

Para-meter	Simulation	Real experiments
$\mathbf{Q}_{FL}$	$\begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$
β	3	4

Table 6.6: Tuning parameters for NMPC

Parameter	Simulation & real experiments
Q_{NMPC}	$\begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001 \end{bmatrix}$
R_{NMPC}	$\begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$

acceptable to carry on with the experiments, with a notion that before every real experiment, the constant error of the RTK GPS units is determined and corrected.

During the same experiments, the IMU gave average errors between 0.3 and 7.5 degrees, where the high deviations occurred in the first five tests. Tests 6-10 gave average errors of 0.3 to 2.1 degrees. The mean absolute deviation around the mean was between 0.8 and 2.9 degrees, while the mean absolute deviation around the true value was between 6.0 and 7.7 degrees in tests 1-5, in tests 6-10 this was between 0.9 and 2.4 degrees. This is found to be acceptable. To compensate the constant error of the IMU, its value is determined before every real experiment and subsequently a correction is made in the software.

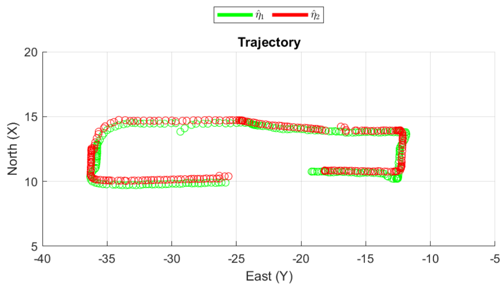
Besides these ten experiments, also an experiment with two Roboat units on top of a moving cart is performed. In this experiment, the distance between the two Roboat units as well as the difference in heading is kept constant, while the absolute position and angle move. The results of this experiment are shown in Figure 6.3. From the results it can be deduced that the relative measured position deviates between -0.70 and 0.55 m from the true value during the test of 150 s. Another result is that a difference in measured heading exists with maximal values of -25 degrees and + 21 degrees. The results show that the error changes relatively smoothly. Since there is no need for the configuration of formation to be very strict and the error changes smoothly it is expected that the inaccuracy of the sensors does not have a big impact for this research. However if the system will be deployed for a specific operation, these errors need to be taken into account.

6.4. Conclusion

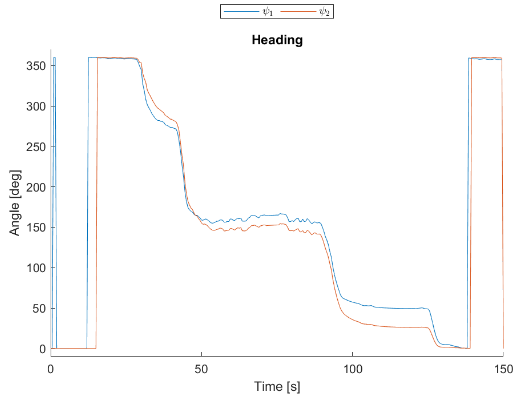
In this chapter, the methods used to evaluate the designed formation control strategies have been explained. During both the simulations and the real experiments, the layout of the test environment is the same. All parameters used in the experiments, have been listed in this chapter. Furthermore, the accuracy of the RTK GPS and IMU sensors have been evaluated. Despite the error of the sensors being larger than expected, it is concluded that the sensors will not influence the results negatively. Based on methods found in existing literature, six simulations and five real experiments will be performed and assessed on five KPIs. Besides, the results of the experiments will be presented in graphs. These results are presented and discussed in the next chapter, Chapter 7.

Table 6.7: An overview of the five real experiments

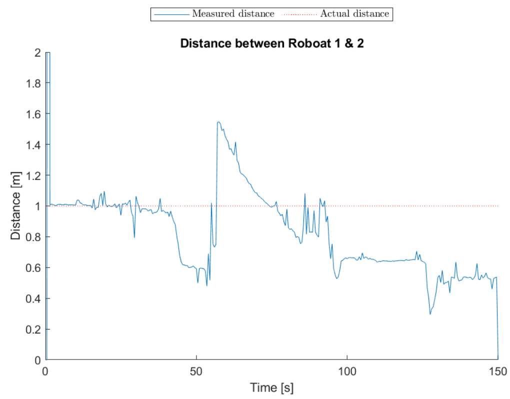
Category	**Parameter**	**Value**				
		Test 2	Test 3	Test 4	Test 5	Test 6
Control system settings	Control method leader	FL MPC	NMPC	PI	FL MPC	NMPC
	Prediction horizon leader (N_p)	20 [-]	10 [-]	-	20 [-]	10 [-]
	Control method follower	FL MPC	NMPC	FL MPC	FL MPC	NMPC
	Prediction horizon follower (N_p)	20 [-]	10 [-]	20 [-]	20 [-]	10 [-]
Guidance system settings	Guidance system leader	complete	complete	global trajectory only	global trajectory only	global trajectory only
	Guidance system followers	complete	complete	complete	complete	complete



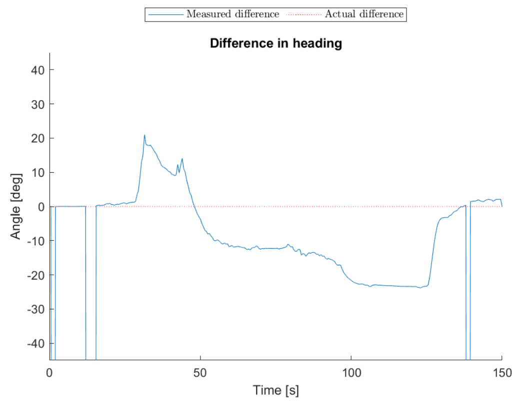
(a) The sailed trajectory



(b) Heading over time



(c) Actual and measured distances between the Roboat units



(d) Difference in heading of the two Roboat units over time

Figure 6.3: Results of the sensor test with two Roboat units

Table 6.8: Parameters used in all tests

Category	**Parameter**	**Value**
Control system settings	Sampling time (t_s)	0.5 [s]
	Minimum force (τ_{min})[a]	$\begin{bmatrix} -4 \\ -4 \\ -2 \end{bmatrix} \begin{matrix} \text{[N]} \\ \text{[N]} \\ \text{[Nm]} \end{matrix}$
	Maximum force (τ_{max})[a]	$\begin{bmatrix} 4 \\ 4 \\ 2 \end{bmatrix} \begin{matrix} \text{[N]} \\ \text{[N]} \\ \text{[Nm]} \end{matrix}$
Thrust allocation settings	Penalty matrix 1 ($\mathbf{P}_{TA}$)[a]	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$
	Penalty matrix 2 ($\mathbf{Q}_{TA}$)[a]	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
	Minimum thrust ($\mathbf{u}_{\text{min}}$)	$\begin{bmatrix} -2 & -2 & -2 & -2 \end{bmatrix}^{\text{T}}$ [N]
	Maximum thrust ($\mathbf{u}_{\text{max}}$)	$\begin{bmatrix} 2 & 2 & 2 & 2 \end{bmatrix}^{\text{T}}$ [N]
	Minimum thrust change ($\Delta\mathbf{u}_{\text{min}}$)	$\begin{bmatrix} -1 & -1 & -1 & -1 \end{bmatrix}^{\text{T}}$ [Ns^{-1}]
	Maximum thrust change ($\Delta\mathbf{u}_{\text{max}}$)	$\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^{\text{T}}$ [Ns^{-1}]
Guidance system settings	Safety distance (d_{safe})	1.5 [m]
	Maximum speed (v_{max})	1.0 [ms^{-1}]
	Headway Roboat 2 (t_{12})	5 [s]
	Headway Roboat 3 (t_{13})	5 [s]
	Lateral distance Roboat 2 (y_{12})	−2 [m]
	Lateral distance Roboat 3 (y_{13})	2 [m]
Input	Waypoints	$\begin{pmatrix} -5 & -5 \end{pmatrix}$ $\begin{pmatrix} -15 & -5 \end{pmatrix}$ $\begin{pmatrix} -15 & -25 \end{pmatrix}$ $\begin{pmatrix} -25 & -25 \end{pmatrix}$ $\begin{pmatrix} -25 & -5 \end{pmatrix}$ $\begin{pmatrix} -15 & -5 \end{pmatrix}$ $\begin{pmatrix} -15 & -25 \end{pmatrix}$ $\begin{pmatrix} -5 & -25 \end{pmatrix}$ $\begin{pmatrix} -5 & -5 \end{pmatrix}$
	Reference speed (v_d)	0.2 [ms^{-1}]
	Duration (T)	300 [s]

[a] Does not apply to NMPC method

7

Results

This chapter is the second part of the experimental assessment of the designed formation control system. Chapter 6 described the evaluation methods of the designed formation control strategy. In this chapter the results of the simulations and real experiments are shown. This will provide an answer to the research question *'What is the performance of the designed formation control strategy and its implementation in Amsterdam's canal network?'*. First, the results of the simulations and afterwards the results of the real experiments will be discussed. Finally, conclusions are drawn based on these results and a short discussion will also be provided.

7.1. Simulation results

As discussed in the previous chapter, six tests have been performed in simulation. In this section the results of these tests are presented in the form of graphs and KPIs as determined in the previous chapter. Per test the following graphs are shown:

1. **Sailed trajectory**: The sailed trajectories of all Roboat units together with the global reference trajectory of Roboat 1 and canal boundaries are plotted;
2. **Position over time and surge speed**: In four graphs the heading, x coordinate, y coordinate, and surge speed of all Roboat units are plotted. In the first three graphs the global reference trajectory of Roboat 1 is added as well;
3. **Distances to other vessels**: In this figure the 'no-collision-constraint' is examined in three graphs that show the distance between the Roboat units over time;
4. **Control input**: In four graphs the control input per Roboat is shown.

In the end of this section, an overview of the five KPIs are given in Figures 7.7 and 7.8. The KPIs that have been discussed in the previous chapter are the following:

1. RMSE of the reference trajectory relative to the global reference trajectory;
2. RMSE of the measured position relative to the reference trajectory;
3. RMSE of the measured heading relative to the reference heading;
4. Square root of total sum of forces applied by each thruster squared;
5. Square root of total sum of the differences in successive control inputs of each thruster squared.

From the six tests that are performed, the leader Roboat has a complete guidance system in the first three tests. In the last three tests, the leader Roboat is equipped with a partial guidance system, meaning that the guidance system only consists of the global trajectory planner (see Section 3.4). The followers have a complete guidance system in all tests. The results of the tests are shown in the following figures:

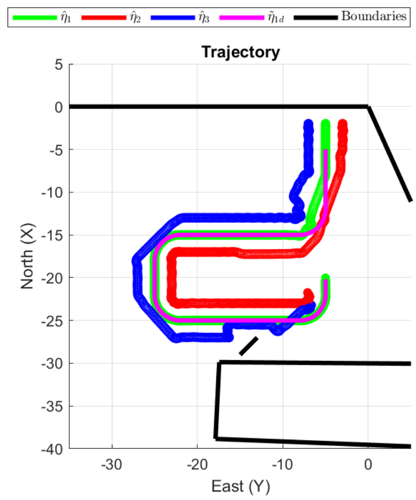
Figure 7.1 Simulation test 1: PI & complete guidance system
Figure 7.2 Simulation test 2: FL MPC & complete guidance system

The initial positions of the three Roboat units in simulation are the following: $\eta_1(0) = [-2, -5, \pi]^{\mathrm{T}}$; $\eta_2(0) = [-2, -3, \pi]^{\mathrm{T}}$; and $\eta_3(0) = [-2, -7, \pi]^{\mathrm{T}}$. The first global reference position of the leader Roboat is $\eta_{d1}(0) = [-5, -5, \pi]^{\mathrm{T}}$. In the next paragraphs, all six tests that have been performed in simulation are analyzed.

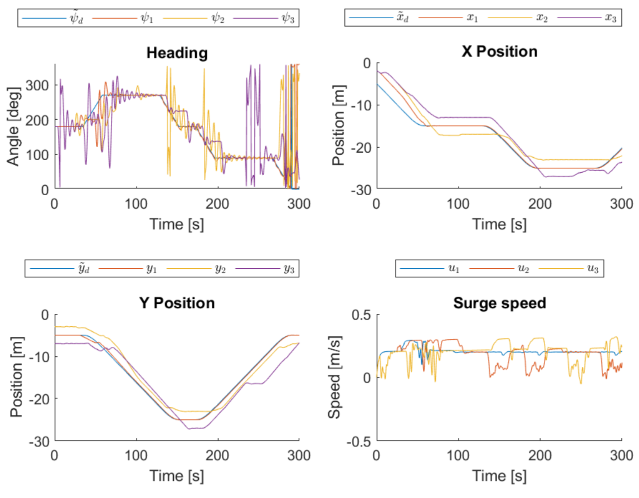
7.1.1. Test 1: PI & complete guidance system

In the first simulation test, the formation control strategy based on the PI control system and complete guidance system for all Roboat units is tested. The parameters used can be found in the previous chapter. Figures 7.1a and 7.1b show that the leader Roboat has difficulties to track the trajectory in the first 60 s. This can be clearly seen in the differences between the KPIs of the whole test and after the first 60 s in the first graph of Figures 7.7 and 7.8, placed in the end of this section. After the initialization phase, the leader Roboat tracks the trajectory accurately, but the heading oscillates slightly. The control input of the leader is steady and changes only slightly, as illustrated in Figure 7.1d. This is also indicated by the KPIs, shown in Figures 7.7 and 7.8.

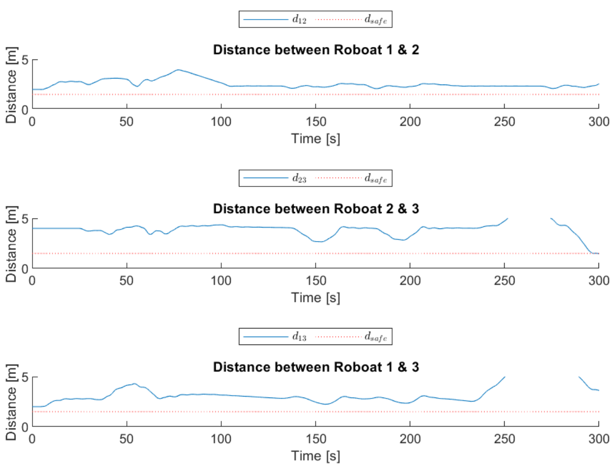
The followers have more trouble in following their global trajectory. Figure 7.1a clearly shows that Roboat 3 is cutting corners. On the other hand, the object is avoided, while maintaining the safety distance with the other Roboat units, as presented in Figure 7.1c. The control input of the follower Roboat units fluctuates fairly, as can be seen in Figure 7.1d and the KPIs, shown in Figures 7.7 and 7.8.



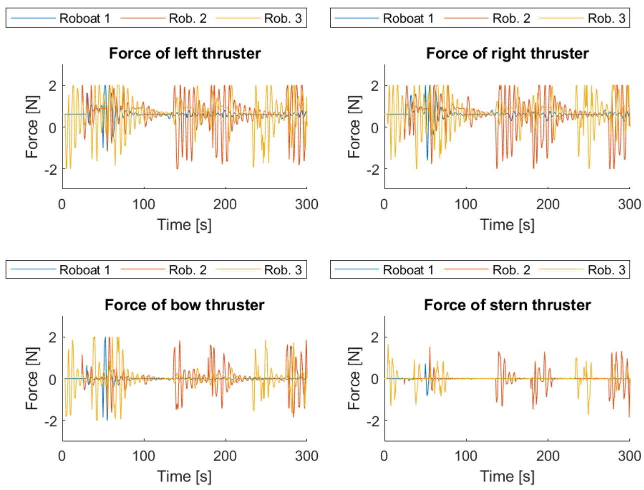
(a) Sailed trajectory



(b) Position over time and surge speed



(c) Distances to other vessels



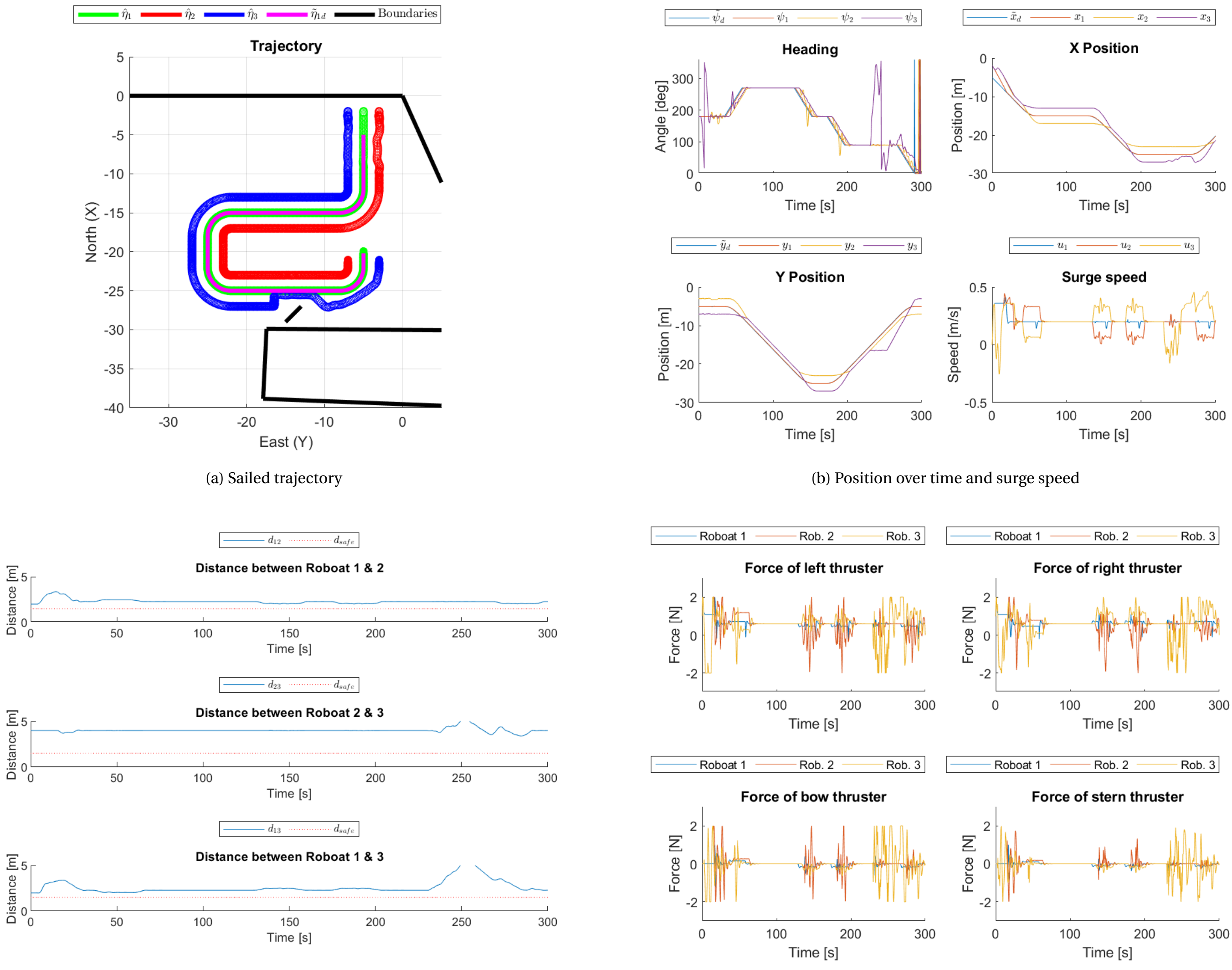
(d) Control input over time

Figure 7.1: Results of simulation test 1: PI & complete guidance system

7.1.2. Test 2: FL MPC & complete guidance system

During the second simulation test, the FL MPC control system, together with the complete guidance system of all Roboat units, forms the formation control strategy. Figures 7.2a and 7.2b show that the leader follows the trajectory correctly. The heading does not oscillate, the surge speed is constant and the control input changes only slightly during the turns, as can be seen in Figure 7.2d. The KPIs, shown in Figures 7.7 and 7.8 confirm this picture.

The trajectories of the follower Roboat units are smooth as well. Roboat 3 cuts the last corner a bit, after avoiding the object. The object is avoided and the distance constraints between the Roboat units are not violated, as can be seen in Figure 7.2c. From Figure 7.2d it strikes that the control inputs of the follower Roboat units change more violently than the control input of the leader Roboat. This is confirmed by the KPIs in Figures 7.7 and 7.8.



(a) Sailed trajectory

(b) Position over time and surge speed

(c) Distances to other vessels

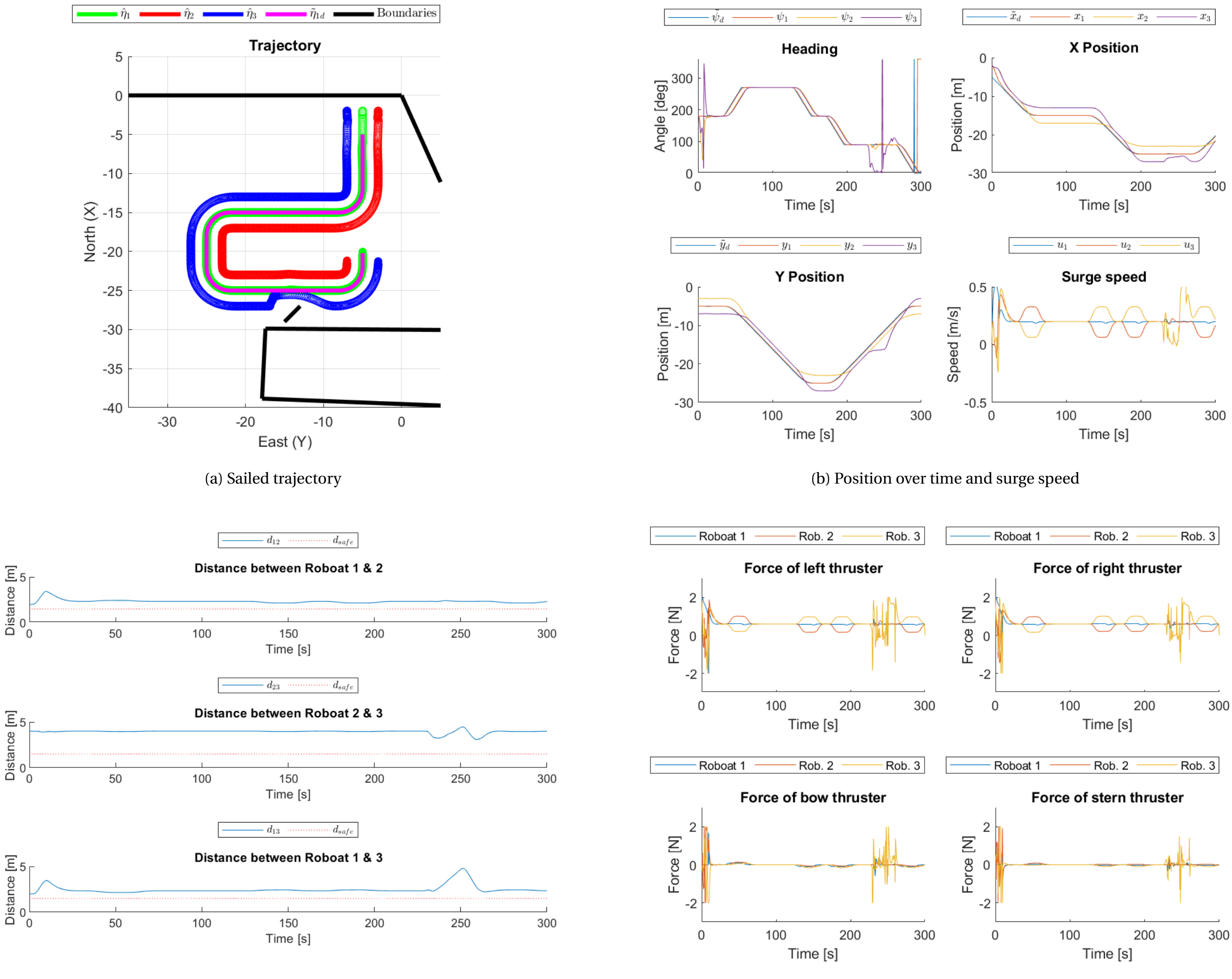
(d) Control input over time

Figure 7.2: Results of simulation test 2: FL MPC & complete guidance system

7.1.3. Test 3: NMPC & complete guidance system

The third simulation test uses the NMPC control system and the complete guidance system for all Roboat units. The global reference trajectory of the leader is followed correctly, as can be seen in Figures 7.3a and 7.3b. The heading does not oscillate, the surge speed is constant as is the control input (Figure 7.3d). This can also be observed in Figures 7.7 and 7.8 that show the KPIs of all simulation tests.

In this test, the follower Roboat units follow their reference trajectory very smoothly as well. The heading does not oscillate at all and Roboat 3 does not cut the last corner. As can be seen in Figure 7.3d, the control input changes frequently only during the first seconds and during the object avoidance of Roboat 3. Furthermore the distance constraints between the Roboat units is clearly not violated, as indicated in Figure 7.3c.



(a) Sailed trajectory

(b) Position over time and surge speed

(c) Distances to other vessels

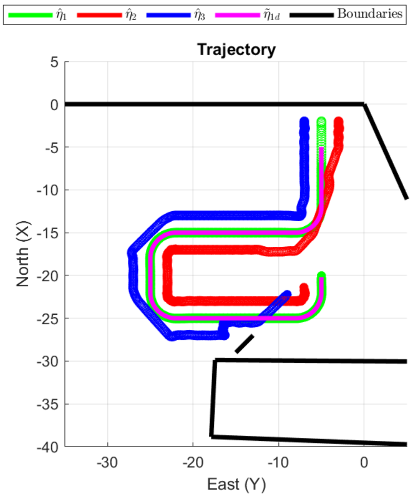
(d) Control input over time

Figure 7.3: Results of simulation test 3: NMPC & complete guidance system

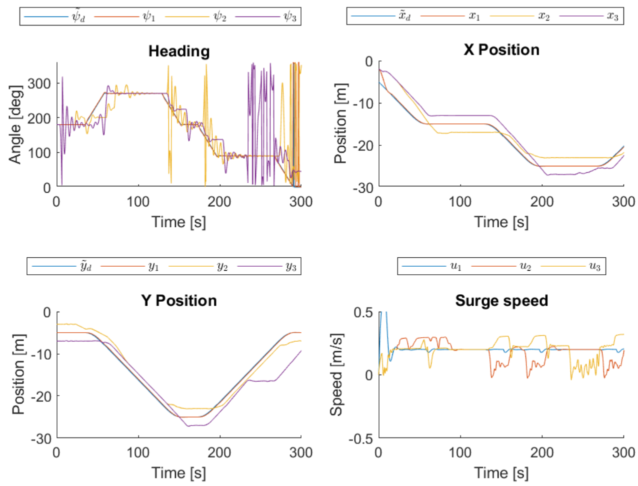
7.1.4. Test 4: PI & partial guidance system leader

In this simulation test, the PI control system is used. The guidance system of the leader creates only a global trajectory. This means that the guidance system of the leader is not influenced by the current position of the vessel and is not able to avoid objects. As shown in Figures 7.4a and 7.4b the leader follows the reference trajectory smoothly, without any oscillations in the heading. The surge speed is constant and Figure 7.13d shows that the control input of the leader does not oscillate as well.

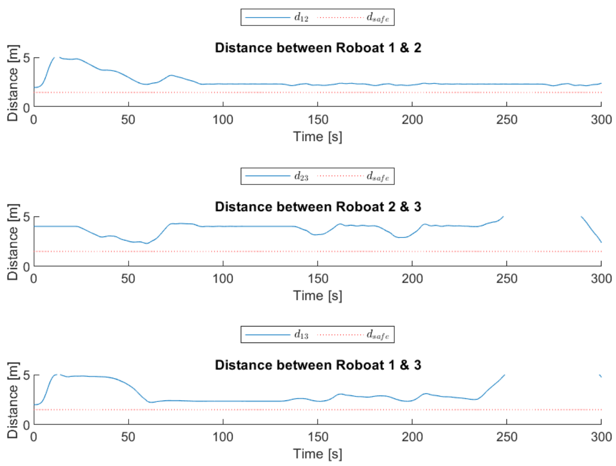
The trajectory tracking of the followers is less smooth. Roboat 2 and 3 cut corners and both the heading and the control input oscillates, as shown in Figures 7.4a, 7.4b and 7.4d. Figure 7.4c shows that the distance constraints have not been violated. All these results are confirmed by the KPIs, shown in Figures 7.7 and 7.8.



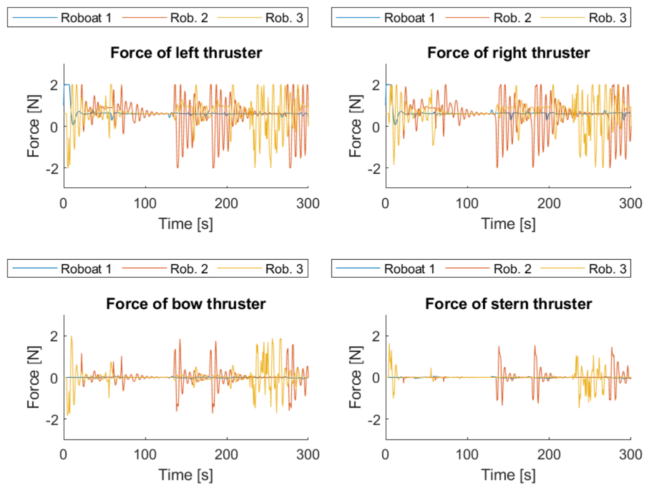
(a) Sailed trajectory



(b) Position over time and surge speed



(c) Distances to other vessels



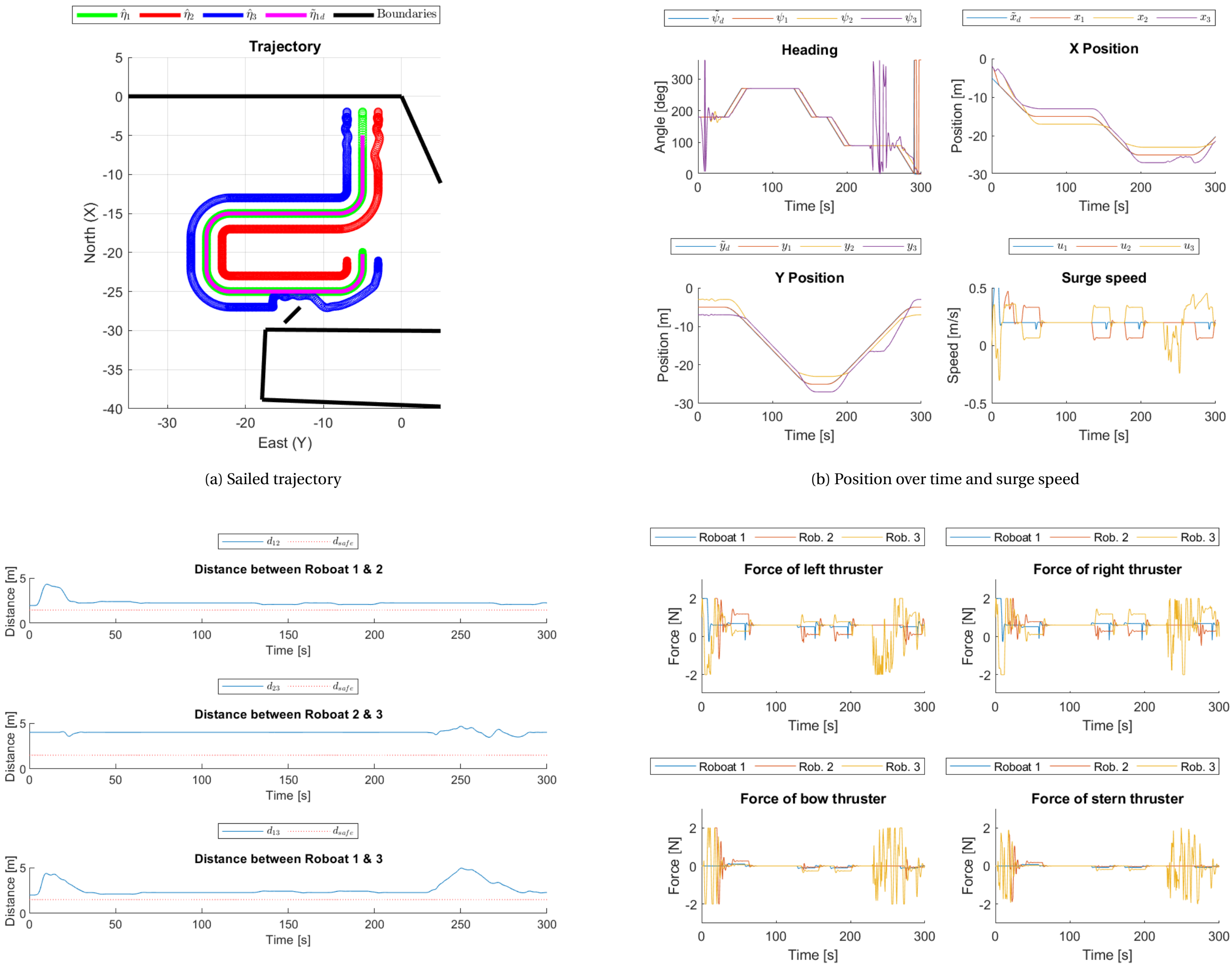
(d) Control input over time

Figure 7.4: Results of simulation test 4: PI & partial guidance system leader

7.1.5. Test 5: FL MPC & partial guidance system leader

The FL MPC control system is used in this test. Furthermore the guidance system of the leader creates only a global trajectory as explained before. The follower Roboat units are equipped with the complete guidance system. Figures 7.5a and 7.5b show that the leader follows its reference trajectory smoothly and without oscillations. The surge speed is constant, as is the control input, shown in Figure 7.14d.

The follower Roboat units track the trajectory good as well. In the beginning and during the object avoidance some oscillations are visible in Figure 7.5b. The speed is quite constant and the trajectory smooth. The distance constraints are not violated as is shown in Figure 7.5c. During the initial phase and the object avoidance, the control inputs are oscillating, but during the steady state phase the control inputs are constant as is shown in Figure 7.5d. All these results are confirmed by the KPIs, shown in Figures 7.7 and7.8 .



(a) Sailed trajectory

(b) Position over time and surge speed

(c) Distances to other vessels

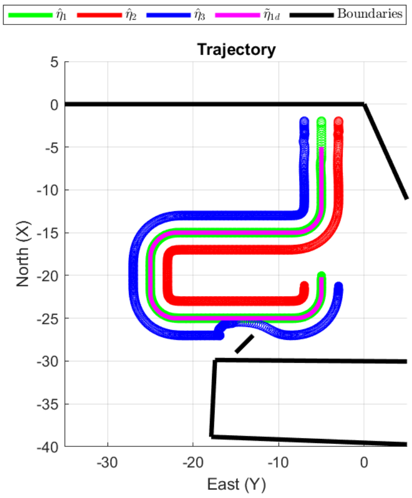
(d) Control input over time

Figure 7.5: Results of simulation test 5: FL MPC & partial guidance system leader

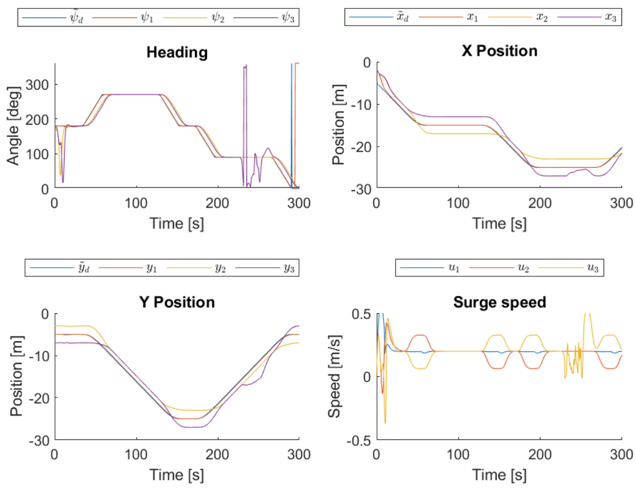
7.1.6. Test 6: NMPC & partial guidance system leader

The final test in simulation makes use of the NMPC control system and for the leader only the global trajectory generator. The followers use the complete guidance system. As shown in Figures 7.6a and 7.6b the leader Roboat follows the reference trajectory accurately. The heading does not oscillate, the surge speed is constant and Figure 7.6d shows that the control input is very constant as well.

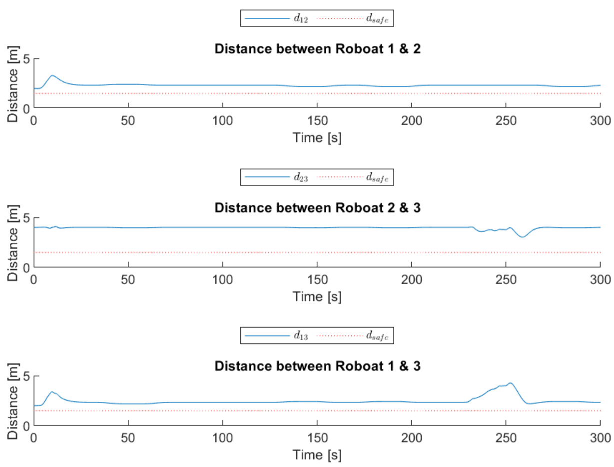
The sailed trajectory of the follower Roboat units is very smooth as well. No corners are cut, the heading does not oscillate and the surge speed is constant. Besides, no distance constraints are violated, indicated by Figure 7.6c. However the control input changes very rapidly during the obstacle avoidance, as can be seen in the Figure 7.6d and in the KPIs, shown in Figures 7.7 and 7.8.



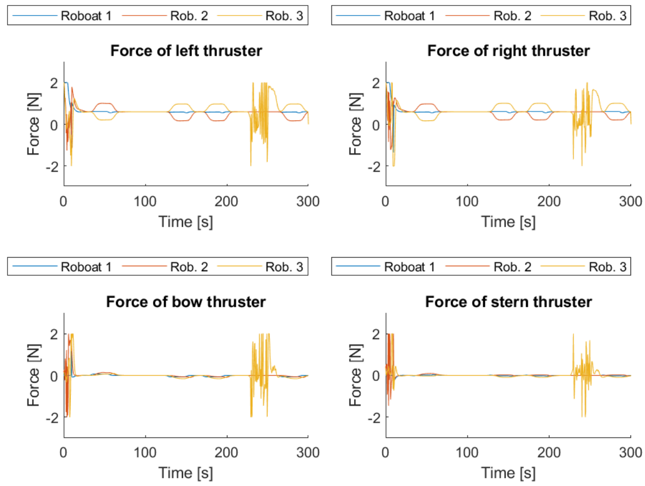
(a) Sailed trajectory



(b) Position over time and surge speed



(c) Distances to other vessels



(d) Control input over time

Figure 7.6: Results of simulation test 6: NMPC & partial guidance system leader

7.1.7. Overview of the KPIs during simulation

In Figure 7.7 and Figure 7.8 overviews of the KPIs during all tests in simulation are given. As discussed in Chapter 6, a difference in behaviour between the steady state and start-up phase often exists in known literature. Figure 7.8 provides the KPIs measured during the complete duration of the tests. In Figure 7.7 the measurement of the KPIs started after the first minute, to exclude the start-up phase. It is encouraging to note that the differences between both figures are not large, since this indicates that the start-up phase went well. To compare the different tests, using the KPIs without start-up phase is most fair. These KPIs are shown in Figure 7.7.

To compare the performance of the three control methods, the KPIs without start-up phase are split in two groups: one group for test 1-3 with complete guidance system and one group for test 4-6 with partial guidance system of the leader Roboat. Subsequently, the results of the KPIs in both groups are normalized to the results of the PI control method: test 1 and test 4. The results of the five KPIs are now compared per group, so every control method is assessed ten times. Most striking is that the NMPC control method outperforms the other control methods seven out of ten times. The other three times, the FL MPC control method performed best. The average of the ten normalized KPIs is 62% for the NMPC control method and 65% for the FL MPC control method. It appears that overall the NMPC control method performs slightly better than the FL MPC control method. Both methods outperform the PI control method significantly. More conclusions, strengthened with the results of the real experiments, can be found further-on in this chapter, in Section 7.3.

The computation time of one control loop is not taken into account in the KPIs, however for the real experiments it is important, as discovered in Chapter 5. The sampling time used in the simulations is 0.5 s, so the control methods that take the dynamics into account assume that the control loop starts every 0.5 s. It is therefore important that the computation time of one control loop is less than 0.5 s. In Figure 7.9 it can be seen that this condition is broken in the period from about 450 s to 500 s, when Roboat 3 has an encounter with the object. In simulations, the real time is not of the essence: the simulation time does not equal the real time and is forced to be 0.5 s during every control loop. In real experiments this is different of course, since the movement of the vessel does not depend on simulation time but on real time. To realize an acceptable computation time, the virtual object is removed during the real experiments.

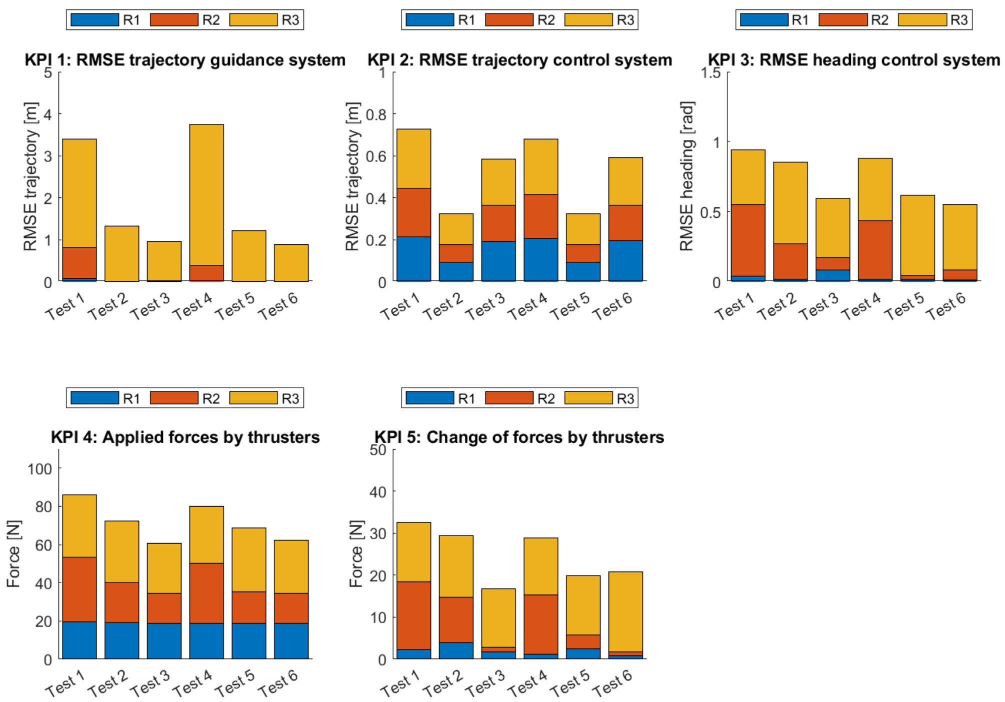


Figure 7.7: KPIs during the six tests in simulation after the initialization phase of one minute (R1 denotes Roboat 1, etc.)

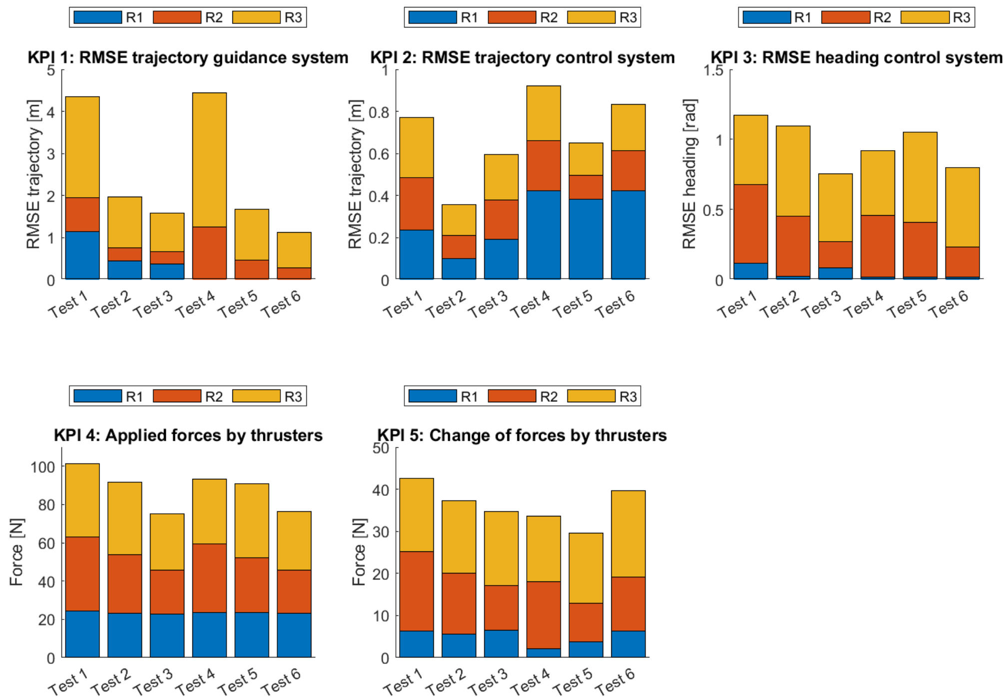
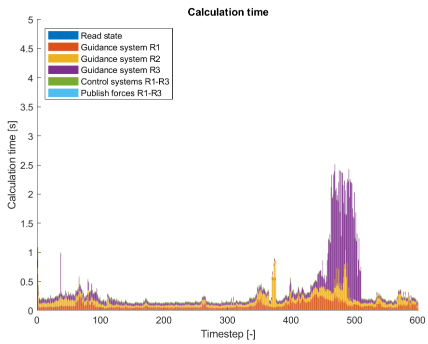
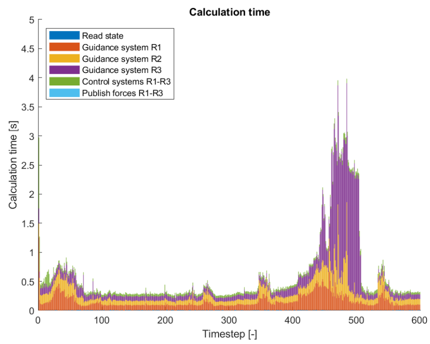


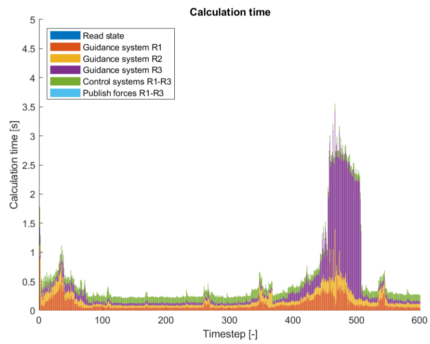
Figure 7.8: KPIs during the six tests in simulation (R1 denotes Roboat 1, etc.)



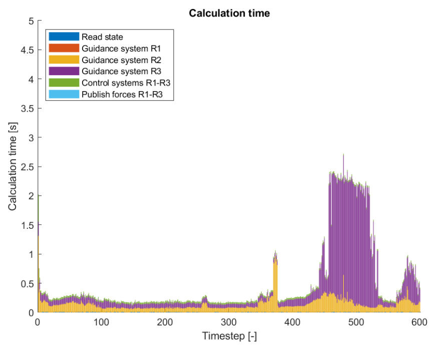
(a) Test 1



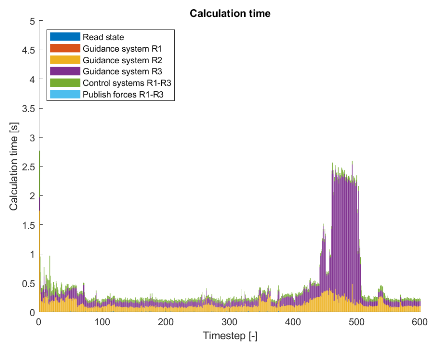
(b) Test 2



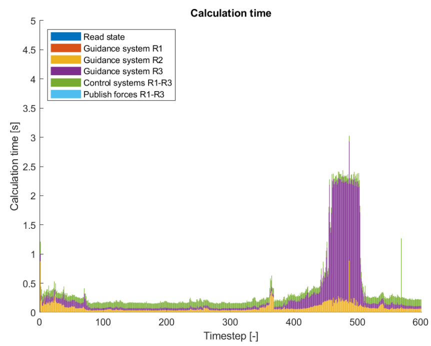
(c) Test 3



(d) Test 4



(e) Test 5



(f) Test 6

Figure 7.9: Computation time per timestep during all simulation tests (R1 denotes Roboat 1, etc.)

7.2. Real experiments results

As explained in Section 6.3.1 the guidance system is found to be incompatible with the PI control strategy in practice. Therefore, the first real experiment is not performed. The five real experiments that are performed in this research, are evaluated in the same way that the simulation tests have been evaluated. During the simulations, the computation time of one time step appeared to be over 2.5 s during the encounter with the object (see Figure 7.9). During the real experiments this is unacceptable, because the sampling time is 0.5 s. Therefore the virtual object is removed during the real experiments.

In the graphs that show the results of the real experiments, the computation time is shown as well. Besides, the real experiments are carried out with only two Roboat units. The results of the real experiments are shown in the following figures:

Figure 7.10 Real experiment 2: FL MPC & complete guidance system
Figure 7.11 Real experiment 3: NMPC & complete guidance system
Figure 7.12 Real experiment 4: PI & half guidance system leader, FL MPC for followers
Figure 7.13 Real experiment 5: FL MPC & half guidance system leader
Figure 7.14 Real experiment 6: NMPC & half guidance system leader
Figure 7.15 KPIs of real experiments without start-up phase
Figure 7.16 KPIs of real experiments

The initial positions of the Roboat units is around $\eta_1(0) = [-2,-5,\pi]^{\mathrm{T}}$; and $\eta_2(0) = [-2,-7,\pi]^{\mathrm{T}}$, but differ per experiment since the Roboat units had to be sailed manually to the initial positions and external disturbances such as waves and wind influenced the position. The first global reference position for the leader Roboat is $\eta_{d1}(0) = [-2,-5,\pi]^{\mathrm{T}}$. In the next paragraphs, all five real experiments are analyzed.

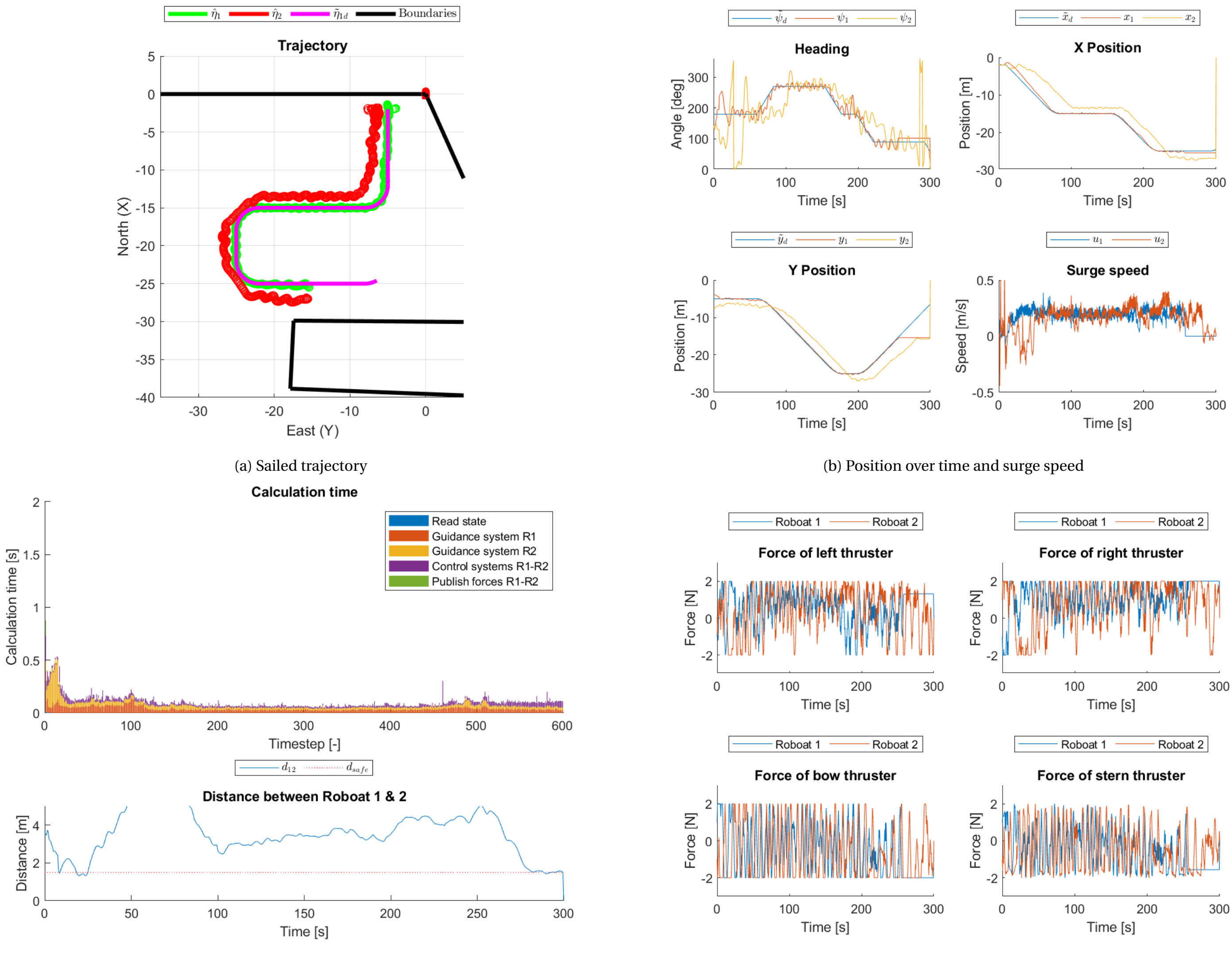
7.2.1. Test 1: PI & complete guidance system (not performed)

The guidance system is found to be incompatible with the PI control strategy in practice, as explained in Section 6.3.1, therefore this test is not performed.

7.2.2. Test 2: FL MPC & complete guidance system

In the first real experiment, the FL MPC control system is used, together with the complete guidance system on both Roboat units. In Figures 7.10a and 7.10b it is shown that the leader follows the trajectory quite good. The heading oscillates slightly, just as the speed. After 260 s, the connection with the leader Roboat is broken, this influences the results of course and can be mainly seen in the first KPI in Figures 7.15 and 7.16. The control input oscillates heavily, as can be seen in Figure 7.10d.

The follower Roboat cuts the corners slightly and its heading oscillates as well. In Figure 7.10c, the distance between the Roboat units is shown. This figure functions only to check whether the distance constraint is broken and therefore only the part of interest of the graph is shown. The distance constraint is just broken in the initial phase by a few centimeters. In the same figure, also the calculation time per timestep is given. It can be seen that the calculation time is below 0.5 s in almost every timestep. Finally, the control input fluctuates in the same manner as that of the leader.



(a) Sailed trajectory

(b) Position over time and surge speed

(c) Distances to other vessels and the calculation time

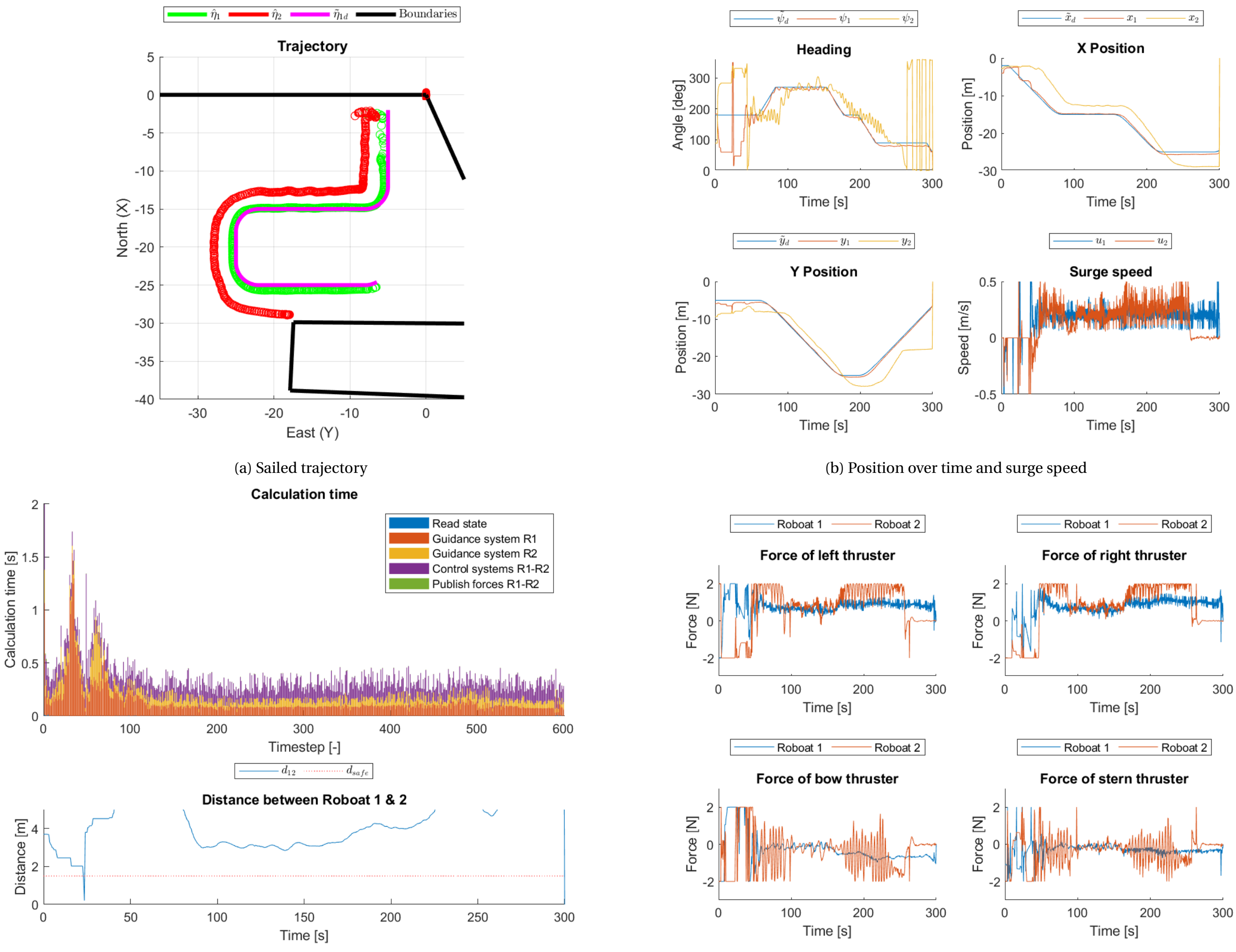
(d) Control input over time

Figure 7.10: Results of real experiment test 2: FL MPC & complete guidance system

7.2.3. Test 3: NMPC & complete guidance system

In this test, the NMPC control system is used, together with the complete guidance system on both Roboat units. Figures 7.11a and 7.11b indicate that the leader Roboat follows the reference trajectory quite accurately, but a slight constant error exists. The heading oscillates slightly in the first phase, but afterwards not that much. The computed surge speed oscillates heavily. Figure 7.11d shows that the control input of the leader is fairly constant.

The heading of the follower Roboat oscillates slightly, but the trajectory is followed quite accurately until 260 s. The follower came to close to an object and is shut down to avoid damage, this explains the high value of the first KPI for Roboat 2 (Figures 7.15 and 7.16). The control input oscillates as shown in Figure 7.11d. Furthermore the distance constraint is violated after 20 s and a small collision occurred (only the part of interest is shown in the graph). The violation of the distance constraint can be explained by the high calculation time during the initial phase, visible in Figure 7.11c. When the calculation time is that high, the system reacts totally different than expected.



(a) Sailed trajectory

(b) Position over time and surge speed

(c) Distances to other vessels and the calculation time

(d) Control input over time

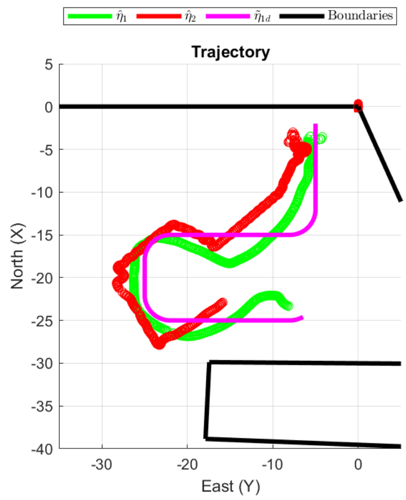
Figure 7.11: Results of real experiment test 3: NMPC & complete guidance system

7.2.4. Test 4: PI & partial guidance system leader

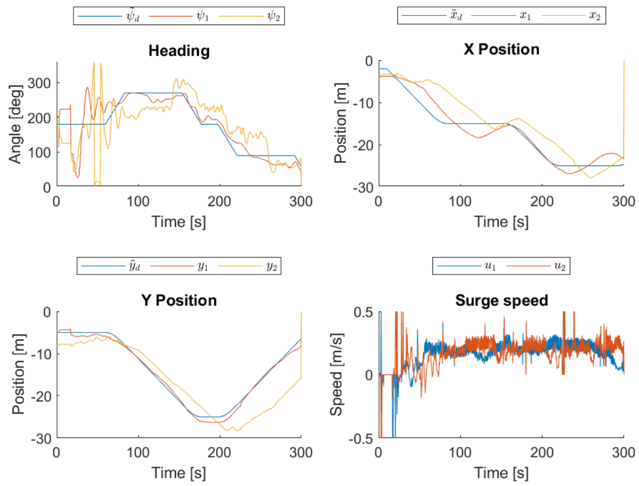
In this real experiment, the set-up of control strategies is different than in the other experiments. The control system of the leader is PI and the guidance system of the leader consists only of the global trajectory planner. The control system of the follower is FL MPC and the follower has a complete guidance system. In the previous chapter it is explained why the complete guidance system and PI did not work together during the real experiments.

From Figures 7.12a and 7.12b can be deduced that the trajectory is not followed correctly by the leader Roboat. The reaction is too slow and overshoots exist, especially in the x coordinate. Figure 7.12d shows that the control input is steady and changes slowly.

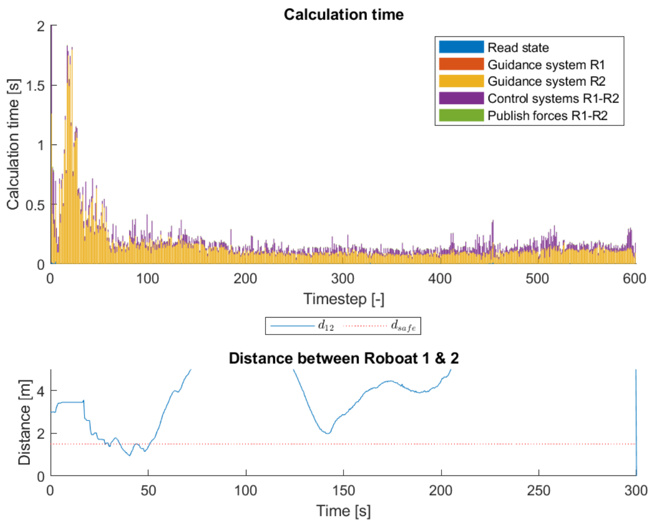
The follower Roboat cuts corners and cannot keep up with the speed of the leader Roboat. The control action is changed frequently and the heading oscillates slightly. Just before the 50 s mark, the distance constraint is violated slightly (in the distance graph, only the area of interest is shown). Figure 7.12c shows that the computation time is above the critical value of 0.5 s in the initial phase. These results are confirmed by the KPIs, shown in Figures 7.15 and 7.16.



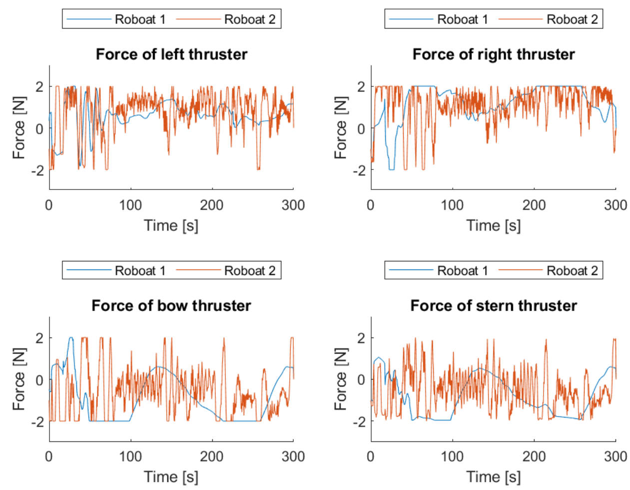
(a) Sailed trajectory



(b) Position over time and surge speed



(c) Distances to other vessels and the calculation time



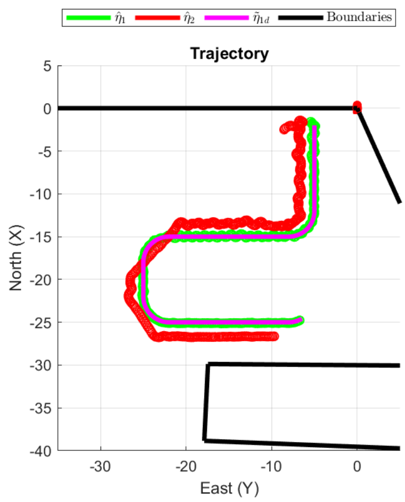
(d) Control input over time

Figure 7.12: Results of real experiment test 4: PI & partial guidance system leader

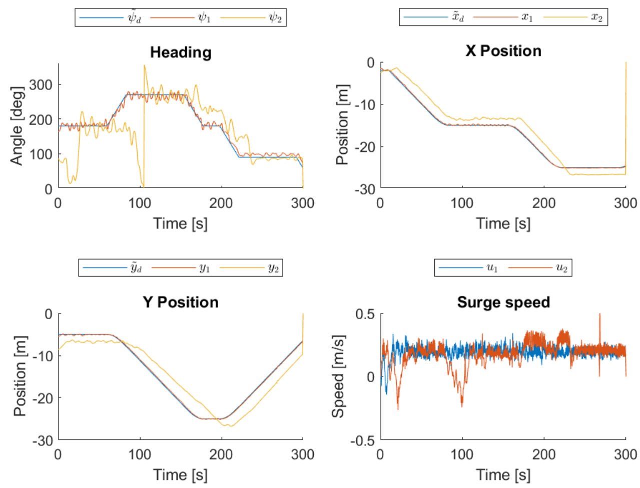
7.2.5. Test 5: FL MPC & partial guidance system leader

In the fifth real experiment, the FL MPC is used as control system. The leader's guidance system consists of only the global trajectory planner, while the follower has a complete guidance system. Figures 7.13a and 7.13b show that the leader tracks the reference trajectory accurately, but both the heading and position oscillate. The surge speed oscillates also. Figure 7.13d shows the frequent changes in control input that probably led to the oscillating position and heading.

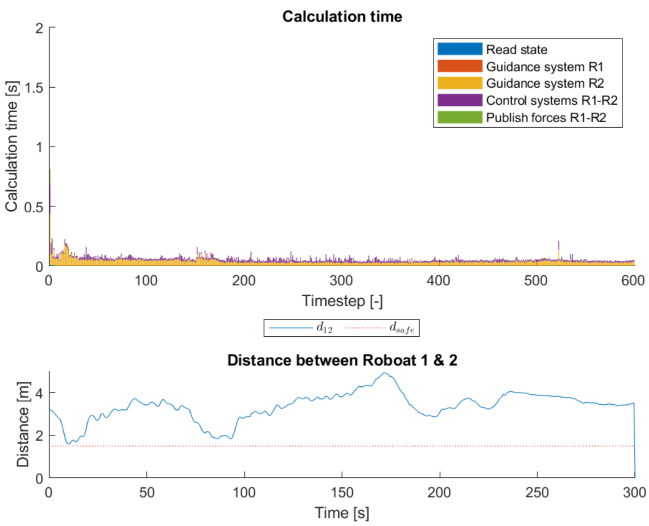
The position of the leader is quite accurately followed by the follower Roboat, but the corners are cut. The heading and surge speed oscillate, just as is the case with the control input. This can be seen in Figures 7.13b and 7.13d. The KPIs, shown in Figures 7.15 and 7.16 confirm the changing behaviour of the control input. Figure 7.13c shows that the distance constraint is not violated and also the calculation time is low. The mentioned results are confirmed by the KPIs, shown in Figures 7.15 and 7.16.



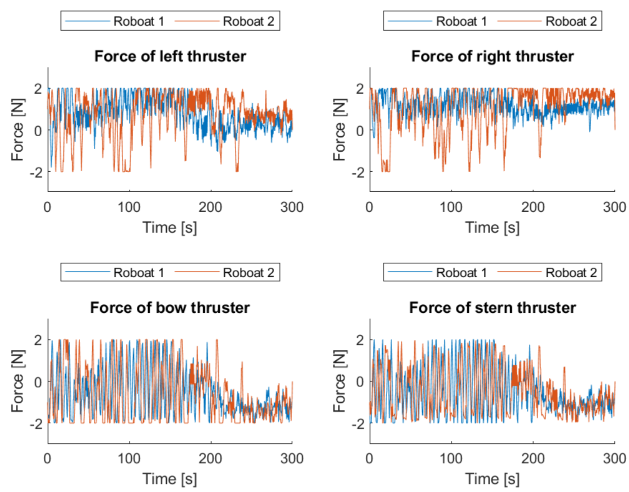
(a) Sailed trajectory



(b) Position over time and surge speed



(c) Distances to other vessels and the calculation time



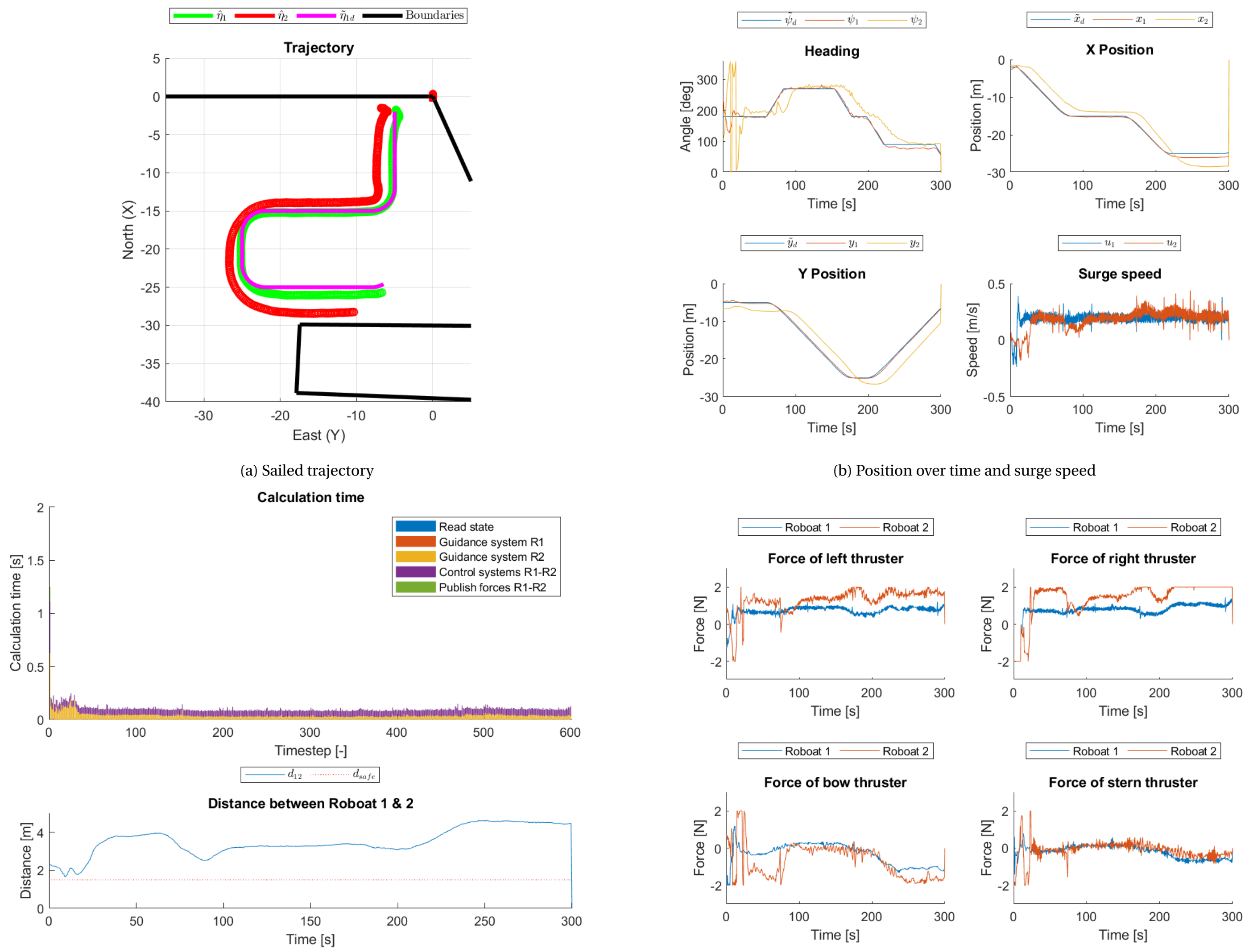
(d) Control input over time

Figure 7.13: Results of real experiment test 5: FL MPC & partial guidance system leader

7.2.6. Test 6: NMPC & partial guidance system leader

In the last real experiment, the NMPC control system is used. The leader's guidance system consists of only the global trajectory planner, while the follower Roboat has a complete guidance system. The leader tracks the reference trajectory accurately without oscillating, as can be seen in Figures 7.14a and 7.14b. Although a small constant error exists in the x coordinate. The speed seems constant, but has a steady oscillation. The control input, shown in Figure 7.14d, is constant, with small changes. This is also clearly visible in the KPIs, shown in Figures 7.15 and 7.16.

The follower Roboat tracks the trajectory smoothly and the heading oscillates barely as shown in Figures 7.14a and 7.14b. The surge speeds does oscillate, but within fixed limits. The control is fairly steady, as shown in Figure 7.14d. In Figure 7.14c, it can be seen that the distance constraint is not violated. Moreover, the calculation time is fairly low throughout the experiment.



(a) Sailed trajectory

(b) Position over time and surge speed

(c) Distances to other vessels and the calculation time

(d) Control input over time

Figure 7.14: Results of real experiment test 6: NMPC & partial guidance system leader

7.2.7. Overview of the KPIs during real experiments

In Figure 7.15 and Figure 7.16 overviews of the KPIs during all real experiments are given. In the same way as the simulation results were presented, a figure with KPIs measured during the complete duration of the tests (Figure 7.16) and a figure with KPIs measured after the first minute to exclude the start-up phase (Figure 7.15) are provided. No remarkable differences between the two figures exist, which is a positive sign, since this indicates that the start-up phase went well. To compare the different tests, using the KPIs without start-up phase is most fair. These KPIs are shown in Figure 7.15.

Comparing the results of the real experiments is harder than comparing the results of the simulation, since many external disturbances act on the Roboat units and these conditions vary throughout the different tests. Furthermore test 1 is not performed as explained in Section 6.3.1. However to compare the performance in the best way possible, the KPIs are split in two groups: one group for tests 2 and 3 with complete guidance system and one group for test 4-6 with partial guidance system. The results of the KPIs in both groups are normalized to the results of the FL MPC control method. The results of the five KPIs are now compared per group, so the PI control method is assessed five times and the other control methods ten times.

In nine out of ten times, the NMPC control method had the best performance, the FL MPC control method performed best one out of ten times. The average value of the normalized KPIs for the NMPC control method is 80%. For the PI control method this is 141%. It can be concluded that the NMPC control method performed better than the FL MPC control method taking all KPIs equally weighted into account. Both the FL MPC and NMPC control method outperform the PI control method significantly. In the next section, these results are combined with the simulation results and more conclusions are drawn.

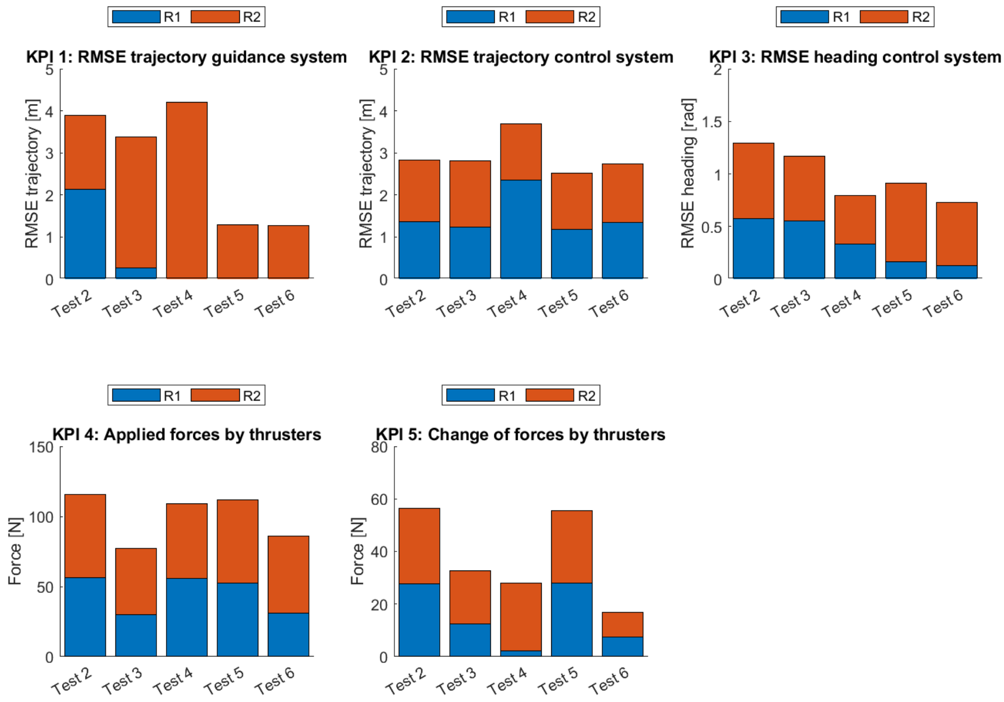


Figure 7.15: KPIs during the five real experiments after the initialization phase of one minute

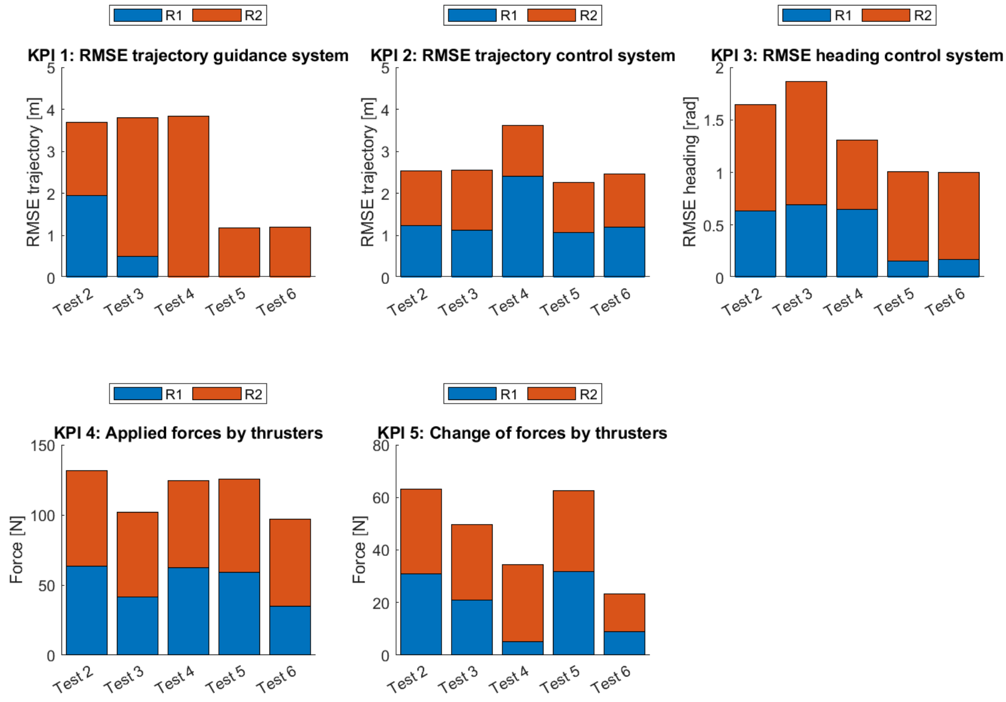


Figure 7.16: KPIs during the five real experiments

7.3. Conclusions of the performed experiments

In this chapter, the results of the simulations and real experiments of this research were presented and analyzed. In this section, the results are compared and conclusions are drawn. Firstly some general conclusions are discussed, then the guidance system and finally the control system are evaluated.

7.3.1. Formation control strategy in general

The first conclusion that can be drawn is that both in simulation and during the real experiments, a flexible formation has been established. In all performed tests, the follower(s) created and followed acceptable trajectories.

The results can be compared with the results of other maritime formation control research works that have been discussed in Section 6.1.2. After the initial phase in simulation the trajectory tracking error of the six tests is barely visible in the figures with position over time, and the maximum error is 0.10 m. Furthermore the error does not oscillate. In comparison with the other research works in simulation that showed errors from 0.05 to 1.0 m, this is very good.

During the real experiments, the trajectory tracking error after the initial phase that can be deduced from the figures, is up to 0.5 m with small oscillation for the FL MPC control method, up to 1.0 m for the NMPC control method and up to 4 m for the PI control method. Compared to the other real experiments that had errors between 0.1 m and oscillating between - 10 and 10 m, this is relatively good.

A second conclusion is that during the simulations, the formation control strategy enabled vessels to avoid objects. During all simulations the object was avoided, while the distance constraints were maintained. However when the object had to be avoided, the computation time of the guidance system became too high to use in practice. For 50 to 100 s, the computation time per timestep went up to 2.5 s, where 0.5 s is the critical value to use in practice. Therefore object avoidance has not been tested during the real experiments.

7.3.2. Guidance system

The guidance system is responsible for determining the reference trajectory. Its performance is measured in three ways. First of all, measured KPI 1, *'The RMSE of the reference trajectory relative to the global trajectory'*, gives an indication on the quality of the guidance system. It is desired to keep this KPI close to zero, since the global reference trajectory is then followed best. This KPI is influenced by the quality of the guidance system, but can also be affected by the control system. Second, the guidance system has been tested by performing tests with and without full guidance system for the leader. Third, the computation time of the guidance system is important for the implementation, therefore this will also be discussed.

KPI 1: RMSE trajectory guidance system During the steady state in simulation, KPI 1 for Roboat 1 and 2 is zero in the tests with FL MPC and NMPC control methods as shown in Figure 7.7. This means that the reference trajectory is equal to the global reference trajectory and is the optimal result. For Roboat 3 this is not the case, as expected since Roboat 3 had to avoid an object. In the simulation tests with the PI control strategy, KPI 1 is higher for Roboat units 2 and 3, with maximum values of 0.73 m for Roboat 2 and 3.34 m for Roboat 3. The larger values of KPI 1 for the PI control strategy are very likely due to the control method. The results of this KPI indicate that the guidance system performs well during simulation, but do not give a definite answer on the performance of the guidance system.

During the real experiments the value of KPI 1 is higher than during the simulations, shown in Figure 7.15. However some values are higher due to specific events. The high value for Roboat 1 during test 1 can be explained by the loss of the connection during the last 40 s, and the high value for Roboat 2 during test 2 can be explained by the shut down of that Roboat, since it came to close to an object. KPI 1 is again higher during the PI test. Apart from these three values, the maximum value of KPI 1 for Roboat 1 and 2 is 1.75 m, belonging to Roboat 2 during the first test.

Tests with and without full guidance system leader Interesting to see is that in simulation not much differences exist in the performance of all KPIs between the first three and last three tests when the first minute is not taken into account, shown in Figure 7.7. Two differences exist: 1) the RMSE of the heading - KPI 3 - for Roboat 2 is 0.25 rad in test 2 whereas it is 0.03 rad in test 5, and 2) the change of control forces - KPI 5 - for Roboat 2 is 10.7 N in test 2 whereas it is 3.3 N in test 5. These two differences are probably related to each other. The lack of differences between the tests with and without full guidance system of the leader provide a strong indication that the guidance system does not influence the performance negatively when no objects are present.

The results of the real experiments with and without full guidance system resemble each other as well, as can be seen in Figure 7.15. In tests 1 and 2 the RMSE of the heading - KPI 3 - of Roboat 1 is higher than in tests 4 and 5. This might indicate that the heading is more constant without the local trajectory planner. Other evidence for this is not gathered. However apart from KPI 3, the results resemble very well and this is another strong indication that the guidance system does not influence the performance negatively when no objects are present. This is very positive, since it is already known that when objects are present, the guidance system influences the performance positively since the objects are avoided.

Computation time A final observation regarding the guidance system is the high computation time. Both in simulation and during the real experiments, most of the computation time was due to the guidance system. Furthermore the reason why object avoidance could not be implemented in the real experiments was the high computation time of the guidance system. The optimization problem of the local trajectory planner is now implemented as MILP problem in MATLAB. In future versions another form of implementation can be chosen to improve the computation time.

Conclusion In summary, KPI 1 gives best results for the FL MPC and NMPC control methods. Without any reference research works that used this KPI it is hard to compare the results with other research works. However the graphs show satisfactory results. Furthermore it can be concluded that the guidance system works less good together with the PI control method. The tests with and without full guidance system of the leader gave comparable results and this indicates that the guidance system works as it should. As final remark, the calculation time has to be lowered in future versions.

7.3.3. Control system

The control system is responsible for calculating the required control inputs to follow the reference trajectory. Its performance is measured by KPI 2-5. Besides, the control system has also influence on KPI 1. In the following paragraphs the three different control systems are compared on the KPIs and the graphs that are shown in the previous chapter.

PI control In simulation the tests with PI control system (test 1 & 4) have the worst performance on almost all KPIs, see Figures 7.7 and 7.8. However, the results seem still acceptable. The RMSE of the measured trajectory relative to the reference trajectory (KPI 2) is less than 29 cm in all cases after the initial phase. Only one real experiment is performed with PI control (test 4). From the graph it is clearly visible that this is the worst control option. The leader Roboat has a large overshoot and has troubles following the reference trajectory. KPI 2 shows an RMSE of the measured trajectory relative to the reference trajectory of 2.36 m. KPI 2 for the other control systems is between 1.19 and 1.37 m.

Comparison FL MPC & NMPC The results of the tests with FL MPC and NMPC control systems are better and more equal. During simulation the NMPC control system (test 3 & 6) scored better on KPI 1, the RMSE of the reference trajectory relative to the global reference trajectory; KPI 3, the RMSE of the measured heading relative to the reference heading; KPI 4, the amount of applied forces; and KPI 5, the change of control input of the thrusters, as presented in Figures 7.7 and 7.8. The FL MPC control method scored better on KPI 2, the RMSE of the measured trajectory relative to the reference trajectory. The results of the real experiments are more equal, but the NMPC control system outperforms the FL MPC control system on KPI 3, 4 and 5, shown in Figures 7.15 and 7.16.

In simulation, the results of the KPIs are normalized to the results of the PI control method. The averages of these normalized KPIs are 62% and 65% for the NMPC and FL MPC control methods respectively. This clearly demonstrates that in simulation the NMPC and FL MPC control methods have similar overall performance - tested with equally weighted KPIs - and both outperform the PI control method. The results of the KPIs of the real experiments are normalized to the values of the FL MPC control method, since only half of the PI control method tests were carried out. The averages of these normalized KPIs are 80% and 141% for the NMPC and PI control methods respectively. This demonstrates that rated on equally weighted KPIs, in practice the NMPC control method performs better than the FL MPC control method. The PI control method yields the worst results of the three control methods.

Computation time The computation time of the control systems during the real experiments can be deduced from the graphs. While the computation times differ each experiment, the clear trend that can be noticed is that FL MPC control algorithms had a lower computation time than the NMPC control algorithms. The results of the tuning experiments, shown in Appendix D, indicate that the computation time of the PI and FL MPC control systems are more or less equal.

The graphs give the indication that the sampling time for the FL MPC control algorithms can be lowered. This would make the accuracy higher.

Conclusion From these results it can be concluded that the NMPC algorithm scored better on most tests. However concerning the low calculation time necessary for the FL MPC algorithm, together with good results, potential for this control algorithm exists as well. The PI control method scored lowest in both the simulation and real experiments.

7.3.4. Discussion on results

Some comments can be made on the results presented earlier and that will be done in the following paragraphs. The first one is that the amount of performed experiments is too low to statistically support the conclusions drawn on the real experiments. During the real experiments many uncertainties exist, such as disturbances due to wind and current, but also delay in communication or inaccuracy of determining the position by RTK GPS due to weather conditions.

Another striking result of the real experiments is the oscillation of the computed surge speed, this also happens when the speed seems constant from the position graph. An explanation is that the calculation of the computed surge speed assumes that the speed at timestep i equals the average speed between timesteps $i-1$ and i. Since the FL MPC control strategy is based on the speed of the vessel, this could have impact on the results of that strategy. In future experiments this could be examined.

The third comment is that the desired speed of the leader Roboat is now set at 0.2 m/s. This is too low in comparison with the desired speed of the full scale Roboat in the canals of Amsterdam. Real experiments with a higher speed were scheduled, but due to current circumstances with Covid-19 these were cancelled. It is expected that due to the increased speed, errors become larger and control parameters need to be optimized again.

On the other hand, during the real experiments, the distributed control architecture is simulated, which has the consequence that on one laptop the control loops for two Roboat units have to be executed in the same amount of time as normally could be done for one Roboat. When the distributed control approach is implemented on the Roboat units, this means that the sampling time can be lowered by a factor of two. Furthermore the control loop will be sped up if implemented in C++ language instead of Matlab. It is expected that the accuracy of the complete system will go up when the sampling time is lowered and thus the control frequency is increased.

In this chapter, the results of six simulations and five real experiments using the designed formation control strategies have been presented. The results are analyzed and compared both with each other and with other research works from literature. The results look promising compared to the research works from literature. The NMPC control method generally gives the best results, followed closely by the FL MPC control method. Both designed methods perform significantly better than the PI control method. In the next chapter, Chapter 8, general conclusions will be drawn. The main findings will be presented and recommendations for future research will be stated.

8
Conclusions, discussion and recommendations

In the previous chapters, a distributed leader-follower formation control strategy is proposed, with three possible control methods. This was done in Chapters 3, 4 and 5, based on the literature review in Chapter 2. Afterwards, the design of the strategies was tested in simulation and during real experiments. The methods for this have been described in Chapter 6 and the results can be found in Chapter 7. In this final chapter the research is concluded and the main findings are presented.

Using the research questions, the main question '*How to design and implement a flexible formation control strategy in Amsterdam's canal network using autonomous Roboat units*' has been answered and in this chapter that will be explained. First of all, the research questions will pass by with short explanations. Thereafter, the scientific relevance of this research is pointed out. The chapter, and with that this research, ends with recommendations for future research.

8.1. Addressing the research questions

Throughout this thesis the research questions composed in Chapter 1 were answered. In this section a retrospect of these questions and way of answering is given. With the answers of the research questions, the main research question has been answered:

How to design and implement a flexible formation control strategy in Amsterdam's canal network using autonomous Roboat units?

8.1.1. What is the current state of research on maritime formation control?

In Chapter 2 a literature review on formation control has been performed. It has been found that in simulation many formation control methods have been proposed. Frequently used control approaches are behavior-based, virtual structures and leader-follower. These approaches have been developed using centralized, decentralized and distributed control.

It is striking that few real experiments have been carried out for maritime formation control. Predictive control methods have not been tested in maritime formation control experiments, but performed experiments with aerial and ground vehicles indicate the large potential for these control methods. The biggest research gap in maritime formation control is found to be the gathering of experimental data in general.

8.1.2. How to design a flexible formation control strategy?

Chapters 3 and 4 gave insights in the design of a distributed leader-follower formation control strategy. In Chapter 3 a control strategy for one Roboat is designed by using a modular approach and dividing the system in four subsystems: marine craft, navigation system, guidance system and control system. The guidance system is designed to make a global trajectory out of user inputs and subsequently create a local trajectory that follows the global trajectory as good as possible, while avoiding objects. Three different control systems have been designed: PI, FL MPC and NMPC.

In Chapter 4 the control strategy is expanded to create a formation of Roboat units. This is done by changing the global trajectory planner in the guidance system of the follower Roboat units.

8.1.3. How to implement the formation control strategy using Roboat units?

In Chapter 5, external disturbances & parameter perturbation, parameter sensitivity, and computation time have been identified as implementation challenges. Using the technical details of Roboat, a command structure is designed. Furthermore, ROS is chosen as communication platform, C++ as programming language for the code on Roboat units, and Matlab as computing environment and programming language for the code of the control system implemented on a laptop on the quay.

8.1.4. What is the performance of the designed formation control strategy and its implementation in Amsterdam's canal network?

The performance of the designed formation control strategy was evaluated in Chapters 6 and 7. The designed control strategies were tested on five KPIs. Besides, graphs with other indicators have been used to graphically show the results of the control strategies. Since many properties of the vessel differ from other vessels used in literature, it is hard to compare the results of the developed control strategies with others. However all formation control strategies developed in this research, provided good results in simulation. The FL MPC and NMPC control strategies performed also good during the real experiments. Due to the high computation time of the guidance system during object avoidance application, this application is not tested during the real experiments. This should be explored in further research.

In simulation the averages of the results of the KPIs normalized to the PI control method are 62% and 65% for the NMPC and FL MPC control methods respectively, where the optimal value of all KPIs is zero. The KPI results of the real experiments are normalized to the results of the FL MPC control method since not all real experiments with the PI control method were carried out. The averages of these normalized results are 80% and 141% for the NMPC and PI control methods respectively. It can be concluded that the overall performance of the FL MPC and NMPC control methods is significantly better than the PI control method, both during simulation and real experiments. While the results of the NMPC and FL MPC control method were similar in simulation, the NMPC control method performs better in practice.

8.1.5. Main question: How to design and implement a flexible formation control strategy in Amsterdam's canal network using autonomous Roboat units?

Throughout this thesis, three flexible formation control strategies were designed and implemented in 1:4 scale Roboat units. The characteristics of Amsterdam's canals were taken into account by including ninety degrees turns in the test trajectory and adding an object in simulation. In the current set-up, the Roboat units are not able to detect other vessels or objects by using their sensors, but thanks to the modular design, this can be added to the current system.

The results of this thesis can be used in the process of transporting household waste via Amsterdam's canals using full scale Roboat units. Throughout this thesis three main reasons for formation control in this specific case were stated:

1. Roboat units give less nuisance to other users of the canals, since they sail close to each other in formation;
2. Roboat units are easier to observe for other vessels and they can be better monitored, leading to an increased level of safety;
3. A probably sooner authorized solution with one human-operated vessel as leader and autonomous Roboat units as followers is revealed.

Besides these advantages for the Roboat project, implementing formation control on the full scale Roboat units would also be a good test case to improve the technology that can be used on larger barges. The contributions are described in detail in the next section.

8.2. Contribution of this research

In this research, formation control strategies for autonomous vessels were investigated and several formation control approaches were proposed, implemented and evaluated. The main contributions of this research can be summarized as follows:

- A literature review on formation control was presented. Using examples from autonomous ground, aerial, underwater, and surface vehicles, the various configuration of formation types, control architectures and control methods have been clearly indicated. Focus is put onto validation of strategies during real experiments, since experimental data lacks in literature.
- Several control strategies for an autonomous vessel were proposed based on modular design. Remarkable parts are the design of the guidance and control system. The proposed guidance system optimizes the reference trajectory to follow a global trajectory as good as possible, while avoiding objects and producing a reachable reference trajectory. Three control systems have been proposed: PI, FL MPC and NMPC control.
- The control strategies for autonomous vessels were extended to create non-iterative distributed flexible leader-follower formation control strategies whereof two control strategies are predictive.
- The designed formation control strategies were implemented on 1:4 scale Roboat units. One of the implemented control strategies includes feedback linearization model predictive control with predictive thrust allocation. To the best knowledge of the authors this has never been tested before in maritime real experiments.
- The designed formation control strategies were validated during simulation and real experiments. To the best knowledge of the author the real experiments with predictive maritime formation control strategies are a first.

8.3. Recommendations for future research

Where possible, the aim of this research was to be as complete as possible. However many research possibilities could not be exploited yet. Besides, throughout the research new research possibilities emerged. In this section, an overview of possible future research opportunities is presented. The recommendations are related to maritime formation control and / or the Roboat project.

1. **Optimal configuration of formation**

   In particular in the road industry, a big advantage of formation control is the reduction of required energy. In maritime formation control, the reduction of friction is more complicated than on the road, since the thrusters of a vessel can also cause negative hydrodynamic effects for a following vessel. More research is desirable to determine the optimal position of the vessels in a formation based on hydrodynamic effects. In Amsterdam's canal network, the limited width of the waterways will also play a big role.

2. **Dynamic model predictive control**

   In this research the weight distribution of the Roboat units is assumed to be constant and known. In practice the Roboat units will transport waste and a logical consequence is that the weight distribution of the vessel or its combination will change. Research to online parameter identification merged with formation control would be very interesting.

3. **Object avoidance in real experiments**

   Due to the high computation time of the guidance system when the reference trajectory had to deviate from the global trajectory because an object got in the way, object avoidance is not tested in the real experiments. However, this is a key element of the designed formation control strategy. In future research, the computation time of the guidance system should be improved, so that object avoidance can be deployed in real experiments

4. **Disturbances estimator**

   In particular during the real experiments, it appeared that disturbances had influence on the performance of the complete system. An addition of (online) estimation of these disturbances to the model used in the MPC strategies would be an improvement of the algorithms.

5. **Optimal coding of designed strategies**

To use the designed strategies on the Roboat units in practical applications, the written code needs to be optimized. The Matlab laptop works inefficiently and should be removed from the system. The algorithms need to be implemented on the mini computers onboard of the Roboat units, to make the system really distributed.

6. **Combining different tasks**

   The designed formation control strategy is very helpful when Roboat units have to sail the same trajectory. However in reality, the the Roboat units will need to split up at a certain point, to perform their assigned tasks. It is recommended that the combination of amongst others the following tasks is researched: entering a formation, formation control, leaving the formation, docking, and latching to dumpsters.

7. **Merging the results of multiple sensors**

   In this research per Roboat one RTK GPS sensor and one IMU were used. To detect objects or other vessels that do not communicate their position, the use of other sensors is required. Research with a Roboat performing SLAM (simultaneous localization and mapping) using a Lidar has already been performed (Mateos et al. (2019)). Fusion of the results of the position determination using an RTK GPS and Lidar would be very interesting and would probably improve the accuracy. Using more sensors could enlarge the reliability of the complete system.

8. **Interaction with human-operated vessels**

   An open and very interesting question is the interaction between fully autonomous and human-operated vessels. The question becomes even more interesting when a formation of autonomous vessels encounters a human-operated vessel. Research opportunities are amongst others: prediction of the behaviour of the human-operated vessel, communication with the human-operated vessel, and desired behaviour of a fully autonomous vessel when approaching a human-operated vessel.

9. **Global operation of waste collection using Roboat units**

   Many knowledge gaps on the complete operation of collecting household waste using Roboat units still exist. Future research should determine which solution is desirable for discarding the household waste in *Amsterdam Centrum*. The current options are: discard the waste in floating dumpsters twice a week, discard the waste in semi-permanent floating dumpsters, or discard the waste in containers built in the quay wall. This will influence the way of transporting the waste with Roboat units. Another research opportunity is to find out how the waste is collected from the Roboat units or dumpsters and how the waste is transported across the waterway *IJ* to the incinerator of *AEB*.

Bibliography

Abdelaal, M., Fränzle, M., and Hahn, A. (2016). NMPC-based Trajectory Tracking and Collison Avoidance of Underactuated Vessels with Elliptical Ship Domain. *IFAC-PapersOnLine*, 49(23):22–27.

Allianz Global Corporate & Specialty (2012). Safety and Shipping 1912-2012. From Titanic to Costa Concordia. page 33.

Alonso-Mora, J., Baker, S., and Rus, D. (2017). Multi-robot formation control and object transport in dynamic environments via constrained optimization. *Int. J. Rob. Res.*, 36(9):1000–1021.

Alonso-Mora, J., Montijano, E., Nägeli, T., Hilliges, O., Schwager, M., and Rus, D. (2018). Distributed multi-robot formation control in dynamic environments. *Auton. Robots*, 43(5):1079–1100.

Alonso-Mora, J., Naegeli, T., Siegwart, R., and Beardsley, P. (2015). Collision avoidance for aerial vehicles in multi-agent scenarios. *Auton. Robots*, 39(1):101–121.

Antonelli, G., Arrichiello, F., and Chiaverini, S. (2009). Experiments of formation control with multirobot systems using the null-space-based behavioral control. *IEEE Trans. Control Syst. Technol.*, 17(5):1173–1182.

Arrichiello, F., Das, J., Heidarsson, H., Pereira, A., Chiaverini, S., and Sukhatme, G. S. (2010). Multi-robot collaboration with range-limited communication: Experiments with two underactuated ASVs. *Springer Tracts Adv. Robot.*, 62:443–453.

AT5 (2019). Megaoperatie vernieuwen kades en bruggen gaat twintig jaar duren.

Avalex (2019). Begroting 2020 en Meerjarenraming 2021-2023.

Bergenhem, C., Pettersson, H., Coelingh, E., Englund, C., Shladover, S., and Tsugawa, S. (2012). Overview of platooning systems. *19th Intell. Transp. Syst. World Congr. ITS 2012*.

Brandstof-zoeker.nl (2020). Adviesprijs van diesel op 9 maart 2020.

Breivik, M., Hovstein, V. E., and Fossen, T. I. (2008). *Ship formation control: A guided leader-follower approach*, volume 17. IFAC.

CBS (2019). Huishoudelijk afval per gemeente per inwoner.

Chen, L., Haseltalab, A., Garofano, V., and Negenborn, R. R. (2019a). ECO-VTF: Fuel-efficient vessel train formations for all-electric autonomous ships. *2019 18th Eur. Control Conf. ECC 2019*, pages 2543–2550.

Chen, L., Hopman, H., and Negenborn, R. R. (2018). Distributed model predictive control for vessel train formations of cooperative multi-vessel systems. *Transp. Res. Part C Emerg. Technol.*, 92(May):101–118.

Chen, L., Hopman, H., and Negenborn, R. R. (2019b). Distributed Model Predictive Control for cooperative floating object transport with multi-vessel systems. *Ocean Eng.*, 191(September).

Chen, L., Huang, Y., Zheng, H., Hopman, H., and Negenborn, R. (2019c). Cooperative Multi-Vessel Systems in Urban Waterway Networks. *IEEE Trans. Intell. Transp. Syst.*, pages 1–14.

Daub, V. (2019). *Roboat : Routing and scheduling of autonomous boats for Amsterdam's waste collection*. PhD thesis, Technische Universitat Darmstadt.

Dean, P., Walters, T., Lequette, S., Goulding, J., Dow, M., and Clack, H. (2018). AUTONOMOUS SHIPS: Shipping 4.0. *HFW Shipp.*, (December):4–7.

Dong, X., Hua, Y., Zhou, Y., Ren, Z., and Zhong, Y. (2019). Theory and Experiment on Formation-Containment Control of Multiple Multirotor Unmanned Aerial Vehicle Systems. *IEEE Trans. Autom. Sci. Eng.*, 16(1):229–240.

ExxonMobil (2015). Hebron Project.

Fleet&Mobility (2018). Bijzonder beheer - Amsterdam introduceert elektrische vuilniswagen.

Fossen, T. I. (2011). Handbook of Marine Craft Hydrodynamics and Motion Control.

Furness, D. (2019). Autonomous ships are coming, and we're not ready for them.

Gemeente Amsterdam (2019). Afvalwijzer Centrum.

Gheneti, B., Park, S., Kelly, R., Meyers, D., Leoni, P., Ratti, C., and Rus, D. (2019). Trajectory Planning for the Shapeshifting of Autonomous Surface Vessels. *IEEE Int. Symp. Multi-Robot Multi-Agent Syst.*

Gheneti, B. H. (2019). Reconfigurable Autonomous Surface Vehicles : Perception and Trajectory Optimization Algorithms. *Master Thesis.*

Ghommem, J., Mnif, F., Poisson, G., and Derbel, N. (2007). Nonlinear formation control of a group of under-actuated ships. *Ocean. 2007 - Eur.*

Guo, K., Li, X., and Xie, L. (2019). Ultra-wideband and Odometry-Based Cooperative Relative Localization With Application to Multi-UAV Formation Control. *IEEE Trans. Cybern.*, pages 1–14.

Haseltalab, A., Garofano, V., Van Pampus, M., and Negenborn, R. (2019). Model Predictive Trajectory Tracking Control and Thrust Allocation for Autonomous Vessels. *Submitted.*

Haseltalab, A. and Negenborn, R. R. (2019). Model predictive maneuvering control and energy management for all-electric autonomous ships. *Appl. Energy*, 251(March).

Haugen, F. (2012). The good gain method for simple experimental tuning of PI controllers. *Model. Identif. Control*, 33(4):141–152.

Houska, B., Ferreau, H., and Diehl, M. (2011). ACADO Toolkit - An Open Source Framework for Automatic Control and Dynamic Optimization. *Optim. Control Appl. Methods*, 32(3):298–312.

Kongsberg Maritime (2020). Autonomous ship project, key facts about YARA Birkeland.

Kuwata, Y. and How, J. P. (2011). Cooperative distributed robust trajectory optimization using receding horizon MILP. *IEEE Trans. Control Syst. Technol.*, 19(2):423–431.

Kyrkjebø, E., Pettersen, K. Y., Wondergem, M., and Nijmeijer, H. (2007). Output synchronization control of ship replenishment operations: Theory and experiments. *Control Eng. Pract.*, 15(6):741–755.

Lawton, J., Beard, R., and Young, B. (2003). A Decentralized Approach to Formation Maneuvers. *IEEE Trans. Robot. Autom.*, 19(6):933–941.

Li, Z., Liu, Z., and Zhang, J. (2020). Multi-under-actuated unmanned surface vessel coordinated path tracking. *Sensors (Switzerland)*, 20(3).

Liang, K. Y., Mårtensson, J., and Johansson, K. H. (2016). Heavy-Duty Vehicle Platoon Formation for Fuel Efficiency. *IEEE Trans. Intell. Transp. Syst.*, 17(4):1051–1061.

Liu, D., Liu, H., Lewis, F. L., and Wan, Y. (2020). Robust fault-tolerant formation control for tail-sitters in aggressive flight mode transitions. *IEEE Trans. Ind. Informatics*, 16(1):299–308.

Liu, Z., Zhang, Y., Yu, X., and Yuan, C. (2016). Unmanned surface vehicles: An overview of developments and challenges. *Annu. Rev. Control*, 41:71–93.

Maestre, J. M. and Negenborn, R. R. E. (2014). *Distributed model predictive control made easy.* Springer Netherlands, 1 edition.

Mahacek, P., Mas, I., Petrovic, O., Acain, J., and Kitts, C. (2009). Cluster space control of autonomous surface vessels. *Mar. Technol. Soc. J.*, 43(1):13–20.

Manoharan, S. H. and Chiu, W. Y. (2019). Consensus Based Formation Control of Automated Guided Vehicles Using Dynamic Destination Approach. *2019 58th Annu. Conf. Soc. Instrum. Control Eng. Japan, SICE 2019*, 902(1):902–907.

Mas, I. and Kitts, C. A. (2014). Dynamic control of mobile multirobot systems: The cluster space formulation. *IEEE Access*, 2:558–570.

Mastellone, S., Stipanović, D. M., Graunke, C. R., Intlekofer, K. A., and Spong, M. W. (2008). Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *Int. J. Rob. Res.*, 27(1):107–126.

Mateos, L. A., Wang, W., Gheneti, B., Duarte, F., Ratti, C., and Rus, D. (2019). Autonomous Latching System for Robotic Boats. *IEEE Int. Conf. Robot. Autom.*, pages 7933–7939.

Mayne, D. Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986.

McCarthy, D. (2014). Choosing between centralized and distributed control system designs.

MI News Network (2018). 7 Major Developments in Autonomous Shipping in 2018.

MUNIN (2016). Maritime Unmanned Navigation through Intelligence in Networks.

Muzaffar, M. (2019). The Future of Maritime Trade? Unmanned Ships.

Negenborn, R. R. and Maestre, J. M. (2014). Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Syst.*, 34(4):87–97.

Novimar (2017). Vessel train concept - Novimar.

Ogren, P. (2004). Split and Join of Vehicle Formations. *Proc. 2004 IEEE Int. Conf. Robot. Autom.*, pages 1951–1955.

OIS Amsterdam (2018). Amsterdam in Cijfers 2018. page 559.

OIS Amsterdam (2019). Kerncijfers Amsterdam 2019. page 40.

Ovcina Mandra, J. (2018). In Depth: Unmanned Ships – Are We There Yet?

Pang, S. K., Li, Y. H., and Yi, H. (2019). Joint formation control with obstacle avoidance of towfish and multiple autonomous underwater vehicles based on graph theory and the null-space-based method. *Sensors (Switzerland)*, 19(11).

Park, S., Kayacan, E., Ratti, C., and Rus, D. (2019). Coordinated Control of a Reconfigurable Multi-Vessel Platform: Robust Control Approach. *IEEE Int. Conf. Robot. Autom.*, pages 4633–4639.

Peng, Z., Gu, N., Zhang, Y., Liu, Y., Wang, D., and Liu, L. (2019). Path-guided time-varying formation control with collision avoidance and connectivity preservation of under-actuated autonomous surface vehicles subject to unknown input gains. *Ocean Eng.*, 191(October).

Perizzato, A., Farina, M., and Scattolini, R. (2015). Formation control and collision avoidance of unicycle robots with distributed predictive control. *IFAC-PapersOnLine*, 23:260–265.

Raghuwaiya, K., Vanualailai, J., and Sharma, B. (2016). Advances in Swarm Intelligence, 7th International Conference, {ICSI} 2016, Bali, Indonesia, June 25-30, 2016, Proceedings, Part {II}. 9713:461–469.

Rashid, M. Z., Yakub, F., Zaki, S. A., Ali, M. S., Mamat, N. M., Mohd Putra, S. M., Roslan, S. A., Shah, H. N., and Aras, M. S. (2019). Comprehensive review on controller for leader-follower robotic system. *Indian J. Geo-Marine Sci.*, 48(7):985–1007.

Riahifard, A., Rostami, S. M. H., Wang, J., and Kim, H. J. (2019). Adaptive leader-follower formation control of under-actuated surface vessels with model uncertainties and input constraints. *Appl. Sci.*, 9(18).

Rijkswaterstaat (1983). Binnenvaartpolitiereglement.

Rolls-Royce Holdings plc (2018). Press releases - Rolls-Royce and Finferries demonstrate world's first Fully Autonomous Ferry.

Schoerling, D., Van Kleeck, C., Fahimi, F., Koch, C. R., Ams, A., and Löber, P. (2010). Experimental test of a robust formation controller for marine unmanned surface vessels. *Auton. Robots*, 28(2):213–230.

Stanford Artificial IntelligenceLaboratory (2018). Robotic Operating System.

Veksler, A., Johansen, T. A., Borrelli, F., and Realfsen, B. (2016). Dynamic Positioning With Model Predictive Control. *IEEE Trans. Control Syst. Technol.*, 24(4):1340–1353.

Wang, J., Gong, S., Peeta, S., and Lu, L. (2019). A real-time deployable model predictive control-based cooperative platooning approach for connected and autonomous vehicles Highlights :. *Elsevier*, 128:271–301.

Wang, W., Mateos, L. A., Park, S., Leoni, P., Gheneti, B., Duarte, F., Ratti, C., and Rus, D. (2018). Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle.

Yu, B., Dong, X., Shi, Z., and Zhong, Y. (2013). Formation control for quadrotor swarm systems: Algorithms and experiments. *Chinese Control Conf. CCC*, pages 7099–7104.

Zhang, Q. and Liu, H. H. T. (2019). Robust Cooperative Formation Control of Fixed-Wing Unmanned Aerial Vehicles. pages 1–18.

Zhang, S., Duarte, F., Guo, X., Johnsen, L., Van de Ketterij, R., and Ratti, C. (2020). On the Feasibility of Using Canals for Waste Collection in Amsterdam. pages 1–29.

Zhang, S., Yang, S., and Xiang, X. (2019). Formation control of autonomous surface vehicle and experimental validation. *IFAC-PapersOnLine*, 52(24):278–282.

Zheng, H., Negenborn, R. R., and Lodewijks, G. (2016). Predictive path following with arrival time awareness for waterborne AGVs. *Transp. Res. Part C Emerg. Technol.*, 70:214–237.

Zheng, H., Wu, J., Wu, W., and Negenborn, R. R. (2019). Cooperative distributed predictive control for collision-free vehicle platoons. *IET Intell. Transp. Syst.*, 13(5):816–824.

Zhu, H., Juhl, J., Ferranti, L., and Alonso-Mora, J. (2019). Distributed Multi-Robot Formation Splitting and Merging in Dynamic Environments. *IEEE Int. Conf. Robot. Autom.*, pages 9080–9086.

A
Scientific research paper

Model Predictive Formation Control for Autonomous Vessels in Inland Waterways⋆

Maurits van Pampus[a,*], Ali Haseltalab[a], Vittorio Garofano[a], Vasso Reppa[a], Ynse Deinema[b], Rudy Negenborn[a]

[a]*Department of Maritime and Transport Technology, Delft University of Technology, Delft, the Netherlands (e-mail: {a.haseltalab,v.garofano,v.reppa,r.r.negenborn}@tudelft.nl)*

[b]*Amsterdam Institute for Advanced Metropolitan Solutions, Amsterdam, the Netherlands (e-mail: ynse.deinema@ams-institute.org)*

Abstract

Formation control of autonomous surface vessels (ASVs) is researched extensively in the last years, since it has promising applications. However very few designed strategies have been validated during real experiments. In this paper, two control methods for distributed leader-follower formation control are proposed: A Nonlinear Model Predictive Control (NMPC) method and an MPC method using Feedback Linearization (FL). These two designed methods are compared with a conventional Proportional-Integral (PI) control method. The performance of the proposed strategies is evaluated through simulations and real experiments using small-scale Roboat units. Both designed control methods outperform the conventional PI control method in simulation and real experiments.

Keywords: Autonomous surface vessels, model predictive control, feedback linearization, formation control

1. Introduction

Formation control is seen as a promising opportunity for autonomous surface vessels (ASVs) and has been studied for years. Advantages of formation control include robustness, reliability and efficiency [1]. Furthermore, formation control is useful for applications such as search and rescue, ocean sampling, large object transport, surveillance, etc [2].

Formation control has been researched using autonomous ground, aerial, underwater and surface vehicles. Most research that is conducted focused on ground and aerial vehicles and control methods used with these vehicles are advanced. For instance NMPC methods have been used in [3] to control a swarm of quadrotors able to avoid objects. An additional reason for formation control with this type of vehicles is fuel savings [4].

Mainly four approaches of formation control have been researched. The behavior-based approach weighs the relative importance of desired prescribed behavior for each vehicle [5, 6]. The virtual structure approach handles the formation as a single entity and assigns a desired motion to the center of the virtual structure [7, 8]. A third method is graph theory that makes use of mathematical structures to model the relations between vehicles [3, 9, 10, 11, 12, 13]. In the leader-follower approach, a designated leader is followed by the followers respecting user-defined distances between the leader and follower [14, 15, 16]. The suitability of each approach depends on the purpose of the formation.

In control theory, three main structures exist: centralized, decentralized and distributed control. While many research works use centralized control [1, 6, 7, 8, 9, 15], it is shown that centralized control does not scale well and is thus less suited for formation control [1, 17, 18]. To cope with these issues decentralized and distributed approaches have been applied. In decentralized control approaches, vehicles do not communicate, but they make their own decisions [5, 14]. In distributed control, vehicles make decisions on their own based on information from other vehicles and from their own sensors [1, 3, 10, 11, 17, 19].

Control methods that have been used for formation control applications are: Proportional-Integral-Derivative (PID) control [5, 6, 9, 15, 20, 21], Linear-Quadratic Regulator [10, 19], Backstepping [7, 14], Neural Network [16], Model Based [8, 22] and Model Predictive Control (MPC) [1, 3, 11, 12, 17, 23, 24, 25, 26, 27]. While MPC is frequently used and validated with simulations, to our best knowledge, no maritime formation control real experiments have been performed with predictive control. The main reason for this seems to be the high computation time. However in other sectors, real experiments have been carried out with MPC [3, 11, 12, 17, 23]. The lack of real experiments with predictive control seems to be a major research gap in maritime formation control.

In this paper, the objective is to design a predictive distributed leader-follower formation control strategy and to implement this in 1:4 scale Roboat units, shown in Figure 1. First, the ship maneuvering model and a Guidance Navigation Control (GNC) structure are briefly explained. Then a guidance system that works based on a Mixed-Integer Linear Programming (MILP) optimization problem is presented. Afterwards, a Nonlinear Model

⋆This research is supported by the Amsterdam Institute for Advanced Metropolitan Solutions (AMS Institute) and by the Researchlab Autonomous Shipping (RAS) at Delft University of Technology

*Corresponding author

(a) 1:4 scale Roboat on the quay in Amsterdam

(b) 1:2 scale Roboat just before going into the water ©2019 AMS Institute

(c) Render of full scale Roboat transporting a dumpster ©2018 Pietro Leoni

Figure 1: The different versions of Roboat

Predictive Control (NMPC) method and an MPC control method using Feedback Linearization (FL) are proposed. A conventional Proportional-Integral (PI) control method is used for comparison. The performance of the various control systems are evaluated using six simulation tests and five real experiments using 1:4 scale Roboat units.

The remainder of this paper is organized as follows. In Section 2 the maneuvering model and GNC structure are presented. In Section 3, the MPC-based approaches are presented. In Section 4 the simulations and real experiments are carried out. Finally, concluding remarks are given in Section 5.

2. System description and problem formulation

In this section, the general maneuvering model of a vessel in 3 Degrees of Freedom (3DoF) is discussed. Afterwards, the general GNC control structure is presented.

2.1. 3DoF maneuvering model

According to [28] the dynamics of a vessel can be generically described by the following nonlinear differential equation:

$$\mathbf{M}\dot{\nu} + \mathbf{C}(\nu)\nu + \mathbf{D}\nu = \tau + \mathbf{d} \tag{1}$$

where $\nu = [u, v, r]^{\mathrm{T}}$ denotes the velocity; $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the added mass and inertia matrix; $\mathbf{C}(\nu) \in \mathbb{R}^{3\times3}$ is the matrix of Coriolis and centripetal terms; $\mathbf{D} \in \mathbb{R}^{3x3}$ is the drag matrix; $\tau \in \mathbb{R}^{3\times1}$ is the vector of body-frame internal forces and moments applied to the vessel; and $\mathbf{d} \in \mathbb{R}^{3\times1}$ the vector of body-frame external forces and moments applied to the vessel. The position and orientation of the ship in the North-East-Down (NED) coordinate system is defined as $\eta = [x, y, \psi]^{\mathrm{T}}$. The kinematic equation relating the velocity components in the NED frame to those in the body fixed frame is:

$$\dot{\eta} = \mathbf{R}(\psi)\nu \tag{2}$$

where $\mathbf{R}(\psi)$ is a Jacobian matrix that transforms a position or velocity vector from the body-fixed to the inertial frame:

$$\mathbf{R}(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Assuming that the origin of the body-fixed frame of the vessel coincides with the center of mass and that the vessel is moving at low speeds, the mass matrix, matrix of Coriolis and centripetal terms and the drag matrix can be described as follows according to [29]:

$$\mathbf{M} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix} \tag{4}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix} \tag{5}$$

$$\mathbf{D} = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \tag{6}$$

Furthermore, τ can be written as function of the control vector $\mathbf{u}$ that contains the forces produced per actuator, using thruster allocation matrix $\mathbf{B}$:

$$\tau = \mathbf{B}\mathbf{u} \tag{7}$$

To obtain a discrete model, the same method as proposed in [2, 30] is used. With state vector $\mathbf{x} = \begin{bmatrix} \eta^{\mathrm{T}} & \nu^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, the dynamic model (1)-(2) can be expressed as:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \tau, \mathbf{d}) \\ &= \begin{bmatrix} \mathbf{0}^{3\times3} & \mathbf{R}(\psi) \\ \mathbf{0}^{3\times3} & \mathbf{M}^{-1}(-\mathbf{C}(\nu) - \mathbf{D}) \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0}^{3\times3} \\ \mathbf{M}^{-1} \end{bmatrix} \tau + \begin{bmatrix} \mathbf{0}^{3\times3} \\ \mathbf{M}^{-1} \end{bmatrix} \mathbf{d} \end{aligned} \tag{8}$$

The dynamic model (8) is now discretized with sampling time t_s to calculate the state at time step $k+1$:

$$\mathbf{x}(k+1|k) = \mathbf{x}(k) + \int_{kt_s}^{(k+1)t_s} f(\mathbf{x}, \tau, \mathbf{d})dt \tag{9}$$

2.2. GNC structure

The control structure used in this research is a guidance, navigation, and control (GNC) structure and schematically shown in Figure 2. The structure consists of four subsystems:

1. **Marine craft**: Includes the vessel itself, the sensors that gather information and the the actuator(s) that apply the control actions to the system;
2. **Navigation system**: Determines the position and velocity of the vessel;
3. **Guidance system**: Responsible for the computation of the reference trajectory;
4. **Control system**: Determines the correct input of the actuators to follow the reference trajectory.

The objective of this research is to design a distributed formation control strategy based on MPC using the GNC structure.

In the next section, the design of the guidance and control system is discussed.

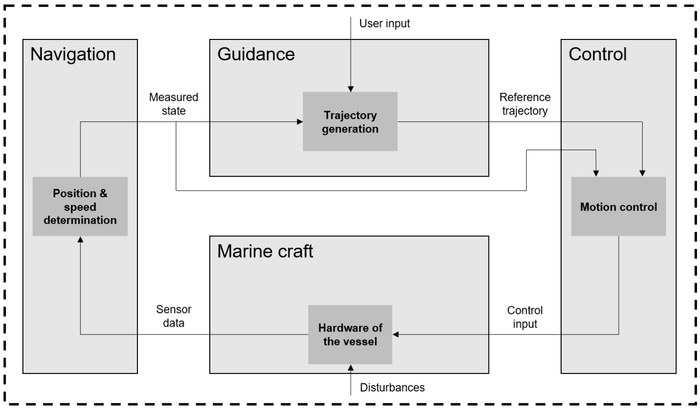


Figure 2: The general control system of a vessel according to [28]

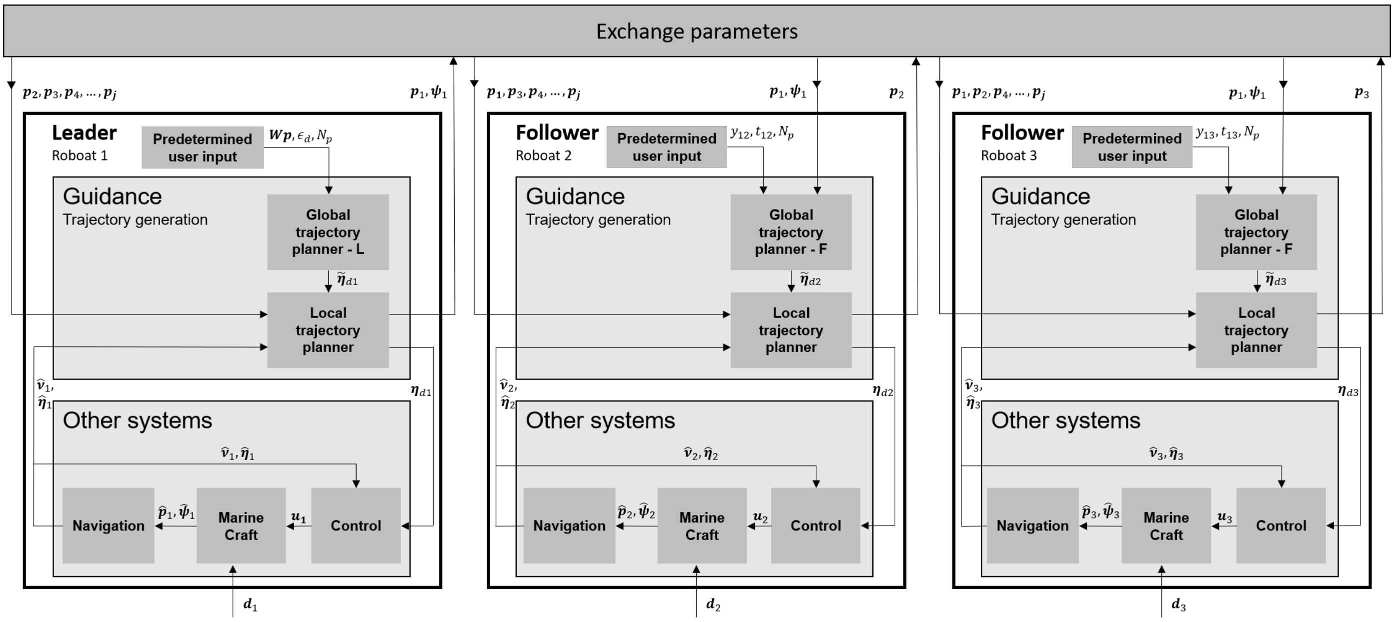
3. Formation control strategy

In this section, a detailed description of the guidance and control system is given. The guidance system is designed in such a way that distributed formation control is established. The global control architecture of the formation is shown in Figure 3. It is shown that positions of other vessels and objects are taken into account in the guidance system. Apart from that, the general control of the vessels resembles the general control shown in Figure 2. In the remaining part of this section, first the proposed guidance system is presented. Afterwards, the proposed control systems are discussed.

3.1. Guidance system

For usage in inland waterways, the guidance system of the leader Roboat must be able to produce a collision free route through the navigable area. Furthermore, the follower Roboat units have to determine their trajectory based on the trajectory of the leader. This is done by creating a *Global trajectory planner* and a *Local trajectory planner*. The global trajectory of the leader is based on user input, while the global trajectory for the followers is based on the predicted and / or sailed trajectory of the leader. The *Local trajectory planner* produces a reference trajectory that minimizes the distance to the global trajectory, while making sure that the trajectory is collision free and inside the navigable area. A systematic method to generate such a trajectory, including static and dynamic constraints, is to use an optimization method. In the following paragraphs the algorithms are explained for the leader and follower Roboat.

Leader Roboat. The *Global trajectory planner* for the leader Roboat has as user inputs: 1) waypoints; 2) the desired surge speed; and 3) the sampling time. With this input, a global reference trajectory is formed.



Declaration of symbols		
$\boldsymbol{p}_j(i)$	$\in \mathbb{R}^{2x1}$	Predicted and current position of vessel (or object) j
y_{1j}	$\in \mathbb{R}$	Desired lateral distance between vessel 1 and j
t_{1j}	$\in \mathbb{R}$	Desired headway between vessel 1 and j
$\hat{\boldsymbol{\eta}}$	$\in \mathbb{R}^{3x1}$	Measured position
$\hat{\boldsymbol{\nu}}$	$\in \mathbb{R}^{3x1}$	Measured velocity
$\boldsymbol{Wp}$	$\in \mathbb{R}^{2xn_{Wp}}$	Waypoints ($n_{Wp} \in \mathbb{N}$; # of waypoints)
ϵ_d	$\in \mathbb{R}$	Reference surge speed
N_p	$\in \mathbb{N}$	Number of prediction steps
$\tilde{\boldsymbol{\eta}}_d(i)$	$\in \mathbb{R}^{3x1}$	Global reference trajectory
$\boldsymbol{\eta}_d(i)$	$\in \mathbb{R}^{3x1}$	Reference trajectory
$\boldsymbol{u}$	$\in \mathbb{R}^{4x1}$	Control input
$\boldsymbol{\tau}$	$\in \mathbb{R}^{3x1}$	Internal forces and moments
$\boldsymbol{d}_j$	$\in \mathbb{R}^{3x1}$	External forces and moments applied to vessel j
$\hat{\boldsymbol{p}}$	$\in \mathbb{R}^{2x1}$	Measured position w/o heading
$\hat{\boldsymbol{\psi}}$	$\in \mathbb{R}$	Measured heading
$\boldsymbol{\psi}_j(i)$	$\in \mathbb{R}$	Predicted and current heading of vessel j

Figure 3: A schematic overview of the control architecture of vessels in formation

The *Local trajectory planner* determines the reference trajectory based on the following optimization problem:

$$J = \min_{\mathbf{p}_1} \sum_{i=0}^{N_p} \|\tilde{\mathbf{p}}_1(k+i) - \mathbf{p}_1(k+i)\| \quad (10)$$

subject to

$$\mathbf{p}_1(k) = \hat{\mathbf{p}}_1(k) \quad (11)$$

$$\|\mathbf{p}_1(k+i) - \mathbf{p}_j(k+i)\| \geq d_{\text{safe}}, \quad i = 0, \ldots, N_p; j = 2, \ldots, m \quad (12)$$

$$v_{\min} \leq \frac{\|\mathbf{p}_1(k+i) - \mathbf{p}_1(k+i-1)\|}{t_s} \leq v_{\max}, \quad i = 1, \ldots, N_p \quad (13)$$

where $\mathbf{p}_1(i) \in \mathbb{R}^{2\times1}$ is the reference position at time step i; $\tilde{\mathbf{p}}_1(i) \in \mathbb{R}^{2\times1}$ is the global reference position; $\hat{\mathbf{p}}_1(k) \in \mathbb{R}^{2\times1}$ is the measured position; $\mathbf{p}_j(i) \in \mathbb{R}^{2\times1}$ is the expected position of object or vessel $j \in \mathbb{N}$, where $j = 1$ is the index for the own vessel and m is the total amount of considered objects and vessels; $d_{\text{safe}} \in \mathbb{R}_{>0}$ is the safety distance between the center of gravity of the own vessel and other vessels or objects; $v_{\min} \in \mathbb{R}$ is the minimum speed; $v_{\max} \in \mathbb{R}$ is the maximum speed; $t_s \in \mathbb{R}_{>0}$ is the sampling time; and $N_p \in \mathbb{N}$ is the number of prediction steps.

The boundaries of inland waterways can be taken into account by regarding them as a sequence of point masses, where every point mass is added to $\mathbf{p}_j$.

This optimization problem is transformed into an MILP problem and solved in MATLAB using the intlinprog solver every time step k. Using the reference position, the reference trajectory is calculated as follows:

$$\psi_d(k+i) = \arctan\left(\frac{y_d(k+i) - y_d(k+i-1)}{x_d(k+i) - x_d(k+i-1)}\right) \quad i = 1, \ldots, N_p; \quad (14)$$

$$\eta_d(k+i) = \begin{pmatrix} \mathbf{p}_1(k+i) \\ \psi_d(k+i) \end{pmatrix} \equiv \begin{pmatrix} x_d(k+i) \\ y_d(k+i) \\ \psi_d(k+i) \end{pmatrix} \quad i = 1, \ldots, N_p; \quad (15)$$

where $\eta_d(i) \in \mathbb{R}^{3\times1}$ is the reference trajectory at time step i, consisting of the components in x, y, and ψ direction: $x_d(i)$, $y_d(i)$, and $\psi_d(i)$.

Follower Roboat. The *Global trajectory planner* for the follower Roboat determines the global reference trajectory based on the trajectory of the leader, a user determined headway, and a user determined desired lateral distance to the leader. This leads to the following equations where

η_1 indicates the position of the leader Roboat and η_j the position of follower Roboat j:

$$n_{1\mathrm{j}} \approx \frac{t_{1\mathrm{j}}}{t_s} \quad (16)$$

$$\eta_1 = \begin{pmatrix} \mathbf{p}_1 \\ \psi_1 \end{pmatrix} \quad (17)$$

$$\tilde{\eta}_d(k+i) = \mathbf{R}\big(\psi_1(a)\big)\Big(\eta_1^b(a) + \begin{pmatrix} 0 \\ y_{1\mathrm{j}} \\ 0 \end{pmatrix}\Big)$$
$$i = 1, \ldots, N_p; \qquad a = k+i-n_{1\mathrm{j}}. \quad (18)$$

The *Local trajectory planner* of the follower Roboat units is the same as that of the leader Roboat.

3.2. Control system

The control system determines the control input of the vessel's thrusters, based on the reference trajectory and measured state. In this paper, we designed two control systems: FL MPC and NMPC. The operation of these systems is discussed in the next paragraphs. The design of the conventional PI controller that is used for comparison is incorporated in Appendix A.

FL MPC. A downside of MPC is the high computation time. To address this problem, the first MPC method that we use, is based on Feedback Linearization [31]. We use the same model as (1) -(2) and (8)-(9) but we do not take disturbances into account, since no adequate solution to determine the disturbances is available in the test set-up. For the sake of Feedback linearization, an auxiliary control input is introduced to establish a linear relationship between the system outputs and auxiliary inputs. Furthermore, the constraints of the optimization problem are linearized, enabling the solution to the problem using quadratic programming methods. This leads to significantly lower computational costs compared to the non linear optimization problem. The FL MPC method is based on the speed dynamics that can be rewritten as:

$$\dot{\nu}(t) = \mathbf{M}^{-1}\Big(\tau(t) - \mathbf{C}(\nu(t))\nu(t) - \mathbf{D}\nu(t)\Big) \quad (19)$$

The following FL control law linearizes the system:

$$\tau(t) = \mathbf{M}\Big(A\nu(t) + B\zeta(t)\Big) + \mathbf{C}(\nu(t))\nu(t) + \mathbf{D}\nu(t) \quad (20)$$

where $\zeta(t) \in \mathbb{R}^{3\times1}$ is the input vector of the linear closed-loop system, $\nu(t)$ is the previously defined velocity vector and represents the states of the linear closed-loop system, and $A \in \mathbb{R}^{3\times3}$ and $B \in \mathbb{R}^{3\times3}$ are states and input matrices of the linear system, respectively. The values of these two matrices can be chosen to the desire of the designer as long as it leads to stability of the created linear closed-loop system.

This system can be discretized in the same way as is done in (9), the new matrices of the state-space representation are now indicated with A_d, B_d. Since this system contains only velocities and no positions, the dynamic system is smaller and thus the computation time is lower. However to accomplish this, the reference position must be transformed to a reference velocity. To this purpose, the reference velocity is approached in two steps. First the average velocity that is required between two time steps is determined. Then this is extrapolated to the required velocity at the second time step, based on the velocity at the first time step and under the assumption that the acceleration is constant:

$$\overline{\nu}_d(k+i|k) = \mathbf{R}^{-1}(\psi(k))\frac{\eta_d(k+i) - \hat{\eta}(k)}{\beta i t_s}$$
$$i = 1, \ldots, N_p; \quad (21)$$

$$\nu_d(k+i|k) = 2\Big(\overline{\nu}_d(k+i|k) - \hat{\nu}(k)\Big) + \hat{\nu}(k)$$
$$i = 1, \ldots, N_p; \quad (22)$$

where $\overline{\nu}_d(k+i|k) \in \mathbb{R}^{3\times1}$ is the average reference velocity of time step $k+i$, calculated at time step k; $\mathbf{R}(\psi(k)) \in \mathbb{R}^{3\times3}$ is the rotation matrix; $\eta_d(k+i) \in \mathbb{R}^{3\times1}$ is the reference position at time step $k+i$; $\hat{\eta}_d(k) \in \mathbb{R}^{3\times1}$ is the measured position at time step k; $\beta \in \mathbb{R}_{>0}$ is a tuning parameter; $t_s \in \mathbb{R}_{>0}$ is the sampling time; $\nu_d(k+i|k) \in \mathbb{R}^{3\times1}$ is the reference velocity of time step $k+i$, calculated at time step k; and $\hat{\nu}(k) \in \mathbb{R}^{3\times1}$ is the measured velocity at time step k. The following optimization problem is now created:

$$J = \min_{\zeta} \sum_{i=1}^{N_p} \Bigg(\Big(\nu(k+i) - \nu_d(k+i)\Big)^{\mathrm{T}} \mathbf{Q}_{\mathrm{FL}}$$
$$\Big(\nu(k+i) - \nu_d(k+i)\Big)\Bigg) \quad (23)$$

subject to

$$\nu(k+i+1) = A_d\nu(k+i) + B_d\zeta(k+i)$$
$$i = 1, \ldots, N_p; \quad (24)$$

$$\tau_{\min} \leq \mathbf{M}\Bigg(A_d\nu(k+i) + B_d\zeta(k+i)\Bigg) + \mathbf{C}\big(\hat{\nu}(k)\big)\hat{\nu}(k) +$$
$$\mathbf{D}\hat{\nu}(k) \leq \tau_{\max} \qquad i = 1, \ldots, N_p; \quad (25)$$

where $\mathbf{Q}_{\mathrm{FL}} \in \mathbb{R}^{3\times3}_{>0}$ is a positive definite penalty matrix. The required forces and moments τ_d can now be approached using the following linearized equation:

$$\tau_d(k+i) = \mathbf{M}\Big(A_d\nu(k+i)+B_d\zeta(k+i)\Big)+\mathbf{C}\big(\hat{\nu}(k)\big)\hat{\nu}(k)+\mathbf{D}\hat{\nu}(k) \tag{26}$$

To address the thrust allocation, an optimization function that minimizes the amount of force and the amount of change of force per thruster is proposed. The proposed thrust allocation algorithm is made predictive, this is especially useful for vessels that have slow reacting actuators. The algorithm is the following:

$$J = \min_{\Delta\mathbf{u}(k)} \sum_{i=0}^{N_{\mathrm{TA}}-1} \Bigg[\Big(\mathbf{u}(k-1)+\sum_{j=0}^{i}\Delta\mathbf{u}(k+j)\Big)^{\mathrm{T}}\mathbf{P}_{\mathrm{TA}} \Big(\mathbf{u}(k-1)+\sum_{j=0}^{i}\Delta\mathbf{u}(k+j)\Big)+ (N_{\mathrm{TA}}-1)\Delta\mathbf{u}(k+i)^{\mathrm{T}}\mathbf{Q}_{\mathrm{TA}}\Delta\mathbf{u}(k+i)\Bigg] \tag{27}$$

subject to

$$\tau_d(k+i) = \mathbf{B}\Big(\mathbf{u}(k+i-1)+\Delta\mathbf{u}(k+i)\Big) \qquad i=1,\dots,N_{\mathrm{TA}}; \tag{28}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(k+i-1)+\Delta\mathbf{u}(k+i) \leq \mathbf{u}_{\max} \qquad i=1,\dots,N_{\mathrm{TA}}; \tag{29}$$

$$\Delta\mathbf{u}_{\min} \leq \Delta\mathbf{u}(k+i) \leq \Delta\mathbf{u}_{\max} \qquad i=1,\dots,N_{\mathrm{TA}}; \tag{30}$$

where $N_{\mathrm{TA}} \in \mathbb{N}$ is the number of prediction steps for the thrust allocation.

NMPC. The second control method makes use of Nonlinear Model Predictive Control (NMPC). The objective function of the NMPC control method is the following:

$$J = \min_{\mathbf{u}(k+i)} \sum_{i=1}^{N_p} \Bigg(\Big(\eta(k+i)-\eta_d(k+i)\Big)^{\mathrm{T}}\mathbf{Q}_{\mathrm{NMPC}} \Big(\eta(k+i)-\eta_d(k+i)\Big)+ \mathbf{u}(k+i)^{\mathrm{T}}\mathbf{R}_{\mathrm{NMPC}}\mathbf{u}(k+i)\Bigg) \tag{31}$$

subject to

$$\mathbf{x}(k+i|k) = \mathbf{x}(k) + \int_{kt_s}^{(k+i)t_s} f(\mathbf{x},\tau)dt \qquad i=1,\dots,N_{\mathrm{P}}; \tag{32}$$

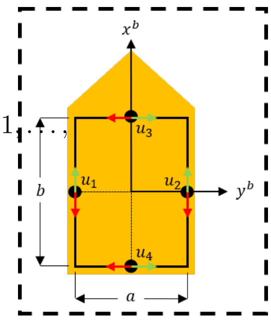


Figure 4: Roboat's thrust configuration. Each thruster is fixed and can generate positive (green) and negative (red) forces ©2020 Maurits van Pampus

$$\mathbf{x}(k+i) = \begin{bmatrix}\eta(k+i)\\ \nu(k+i)\end{bmatrix} \qquad i=1,\dots,N_{\mathrm{P}}; \tag{33}$$

$$\tau_d(k+i) = \mathbf{B}\mathbf{u}(k+i) \qquad i=1,\dots,N_{\mathrm{P}}; \tag{34}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(k+i) \leq \mathbf{u}_{\max} \qquad i=1,\dots,N_{\mathrm{P}}. \tag{35}$$

where $\mathbf{Q}_{\mathrm{NMPC}} \in \mathbb{R}^{3\times3}_{>0}$ and $\mathbf{R}_{\mathrm{NMPC}} \in \mathbb{R}^{3\times3}_{>0}$ are positive definite penalty matrices. As can be noticed, disturbances are not taken into account in the NMPC control method as well, since they cannot be measured.

This control method incorporates the thrust allocation in the optimization problem. It is suggested by [32] that this will result in a more optimal control output compared to a separate thrust allocation system.

4. Simulation and real experiments results

In this section, the performance of the proposed formation control strategy is evaluated during six tests in simulation and five real experiments using 1:4 scale Roboat units. First, the details of the Roboat units are presented. Then, the methodology for the experiments is discussed. Finally, the results of the simulations and real experiments are analysed.

4.1. System description

For both the simulations and the real experiments, the dynamic model of the 1:4 scale Roboat unit is used. The model parameters used, have been determined experimentally in the research of [29] and are shown in Table 1. Other characteristics and properties of the 1:4 scale Roboat units have been listed in Table D.6, shown in Appendix D.

Table 1: Results of the parameter identification conducted by [29].

Item	m_{11}	m_{22}	m_{33}	X_u	Y_v	N_r
Value	12.982	23.318	1.273	6.012	7.112	0.771

The thrust configuration of Roboat is graphically shown in Figure 4 and since the thrust configuration of the Roboat is known, τ can be written as:

$$\tau = \mathbf{B}\mathbf{u} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ a/2 & -a/2 & b/2 & -b/2 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} \quad (36)$$

where $\mathbf{B}$ is the thruster allocation matrix and $\mathbf{u}$ is the control vector. The distance between the transverse propellers is represented by a and the distance between the longitudinal propellers by b, with $a = 0.90$ m, and $b = 0.45$ m. Each propeller is fixed and can generate forward and backward forces.

During the real experiments, the communication platform used is ROS [33], hosted by a laptop using a quad core 1.8 GHz Intel(R) Core(TM) i7-8550U processor with 16 GB of RAM. A central laptop with a simulated distributed control approach runs a MATLAB script that determines the control inputs of the Roboat units, this laptop has the same processor and 8 GB of RAM. The control action is sent to the mini pcs on board the Roboat units that transform the signal to an input signal of the thrusters. Sensor data is continuously sent to the central MATLAB computer. The code on the mini pcs is written in C++. The MATLAB control loop is executed at 2 Hz.

4.2. Methodology

In simulation, the control methods of the first three tests are PI control, FL MPC, and NMPC respectively. The parameters for implementing PI, FL MPC and NMPC are given in Appendix C. During the remaining three tests, the same control methods are used, but the guidance system of the leader Roboat consists of only the global trajectory planner. During the real experiments, the same tests except for Test 1 are performed, and Test 4 (PID control with partial guidance system of the leader) uses FL MPC as control system for the follower Roboat. The FL MPC problem is solved in MATLAB using Quadratic Programming, while the NMPC problem is solved using the ACADO toolkit [34].

A trajectory is sailed in front of the AMS Institute building on the *Marineterrein* in Amsterdam. In simulation an object that has to be avoided is added. During the real experiments, it appeared that the computation time to avoid the object was too high. In simulation we considered three and during the real experiments two Roboat units. One dedicated leader Roboat is followed by the follower(s) with a headway of 5 s and lateral distance of 2 m[1]. The prediction horizon for the FL MPC method is 10 s, for the NMPC method it is 5 s. An overview of all parameters, including the tuning parameters of all control methods is given in Appendix C.

[1]In simulation this is -2 m for one of the follower Roboat units

4.3. Results

During the tests, each vessel is assessed on five KPIs. The results of the simulation, including a short explanation of the KPIs are shown in Figure 5. In Figure 6, the results of the real experiments are shown. A thorough explanation of the calculation of the values of the KPIs can be found in Appendix B.

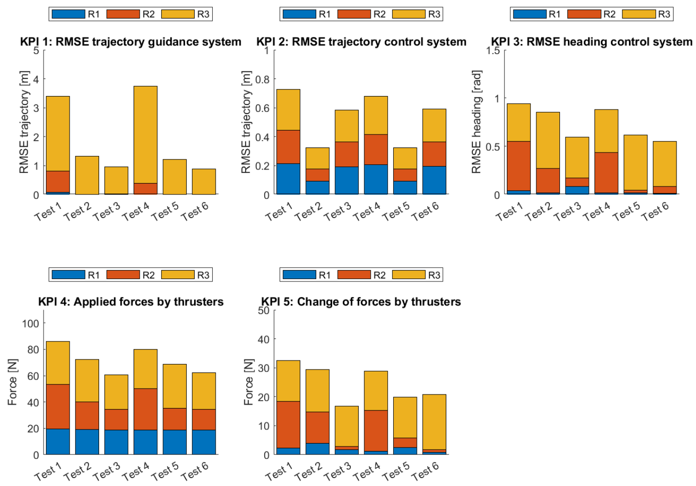


Figure 5: KPIs during the six tests in simulation after the initialization phase of 1 minute

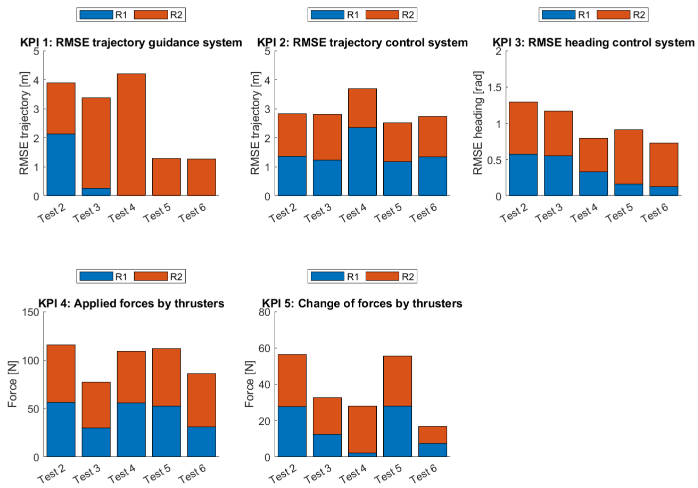


Figure 6: KPIs during the five real experiments after the initialization phase of 1 minute

Valuable insights provided by Figure 5 and Figure 6 are the following. First of all it can be concluded that the tests with and without full guidance system of the leader gave comparable results and this indicates that the guidance system works as it should. Second, it can be seen that the NMPC method scored better than the other methods in most tests. The conventional PI control method scored lowest in both the simulation and real experiments.

Furthermore, the results of the simulation KPIs are normalized to the results of the PI control method. The averages of these normalized KPIs are 62% and 65% for the NMPC and FL MPC control methods respectively. This clearly demonstrates that in simulation the NMPC and FL MPC methods have similar overall performance - tested with equally weighted KPIs - and both outperform the PI control method. The results of the KPIs of the real experiments are normalized to the values of the FL MPC method, since only half of the PI control method tests were carried out. The averages of these normalized KPIs are 80% and 141% for the NMPC and PI control methods respectively. This demonstrates that rated on equally weighted KPIs, in practice the NMPC method performs better than the FL MPC control method. The PI control method yields the worst results of the three control methods.

Detailed results of Test 6, performed both in simulation and real experiments, are added in Appendix E in Figure E.7 and Figure E.8.

5. Conclusion

In this paper, distributed Leader-Follower formation control strategies based on Model Predictive Control have been proposed. A guidance system using an MILP optimization problem that is able to perform object avoidance and three control systems have been designed. Both the NMPC and the FL MPC method gave better results than the conventional PI control method in simulation and during real experiments. The total costs in simulation were 38 and 35 percentage points less for the NMPC and FL MPC methods respectively compared to the PI control method. During the real experiments the total costs were 43 and 29 percentage points less for the NMPC and FL MPC methods respectively. In future research the computation time can be lowered so the frequency of the control loop can be increased and with that the performance of the system.

Other opportunities for future research are amongst other things: determining the optimal configuration of formation based on hydrodynamic effects; dynamic model predictive control to account for different weight distributions; and combining different tasks such as entering a formation, leaving the formation, and docking.

References

[1] L. Chen, H. Hopman, R. R. Negenborn, Distributed model predictive control for vessel train formations of cooperative multi-vessel systems, Transp. Res. Part C Emerg. Technol. 92 (2018) 101–118.

[2] L. Chen, A. Haseltalab, V. Garofano, R. R. Negenborn, ECO-VTF: Fuel-efficient vessel train formations for all-electric autonomous ships, 2019 18th Eur. Control Conf. ECC 2019 (2019) 2543–2550.

[3] H. Zhu, J. Juhl, L. Ferranti, J. Alonso-Mora, Distributed Multi-Robot Formation Splitting and Merging in Dynamic Environments, IEEE Int. Conf. Robot. Autom. (2019) 9080–9086.

[4] K. Y. Liang, J. Mårtensson, K. H. Johansson, Heavy-Duty Vehicle Platoon Formation for Fuel Efficiency, IEEE Trans. Intell. Transp. Syst. 17 (2016) 1051–1061.

[5] J. Lawton, R. Beard, B. Young, A Decentralized Approach to Formation Maneuvers, IEEE Trans. Robot. Autom. 19 (2003) 933–941.

[6] G. Antonelli, F. Arrichiello, S. Chiaverini, Experiments of formation control with multirobot systems using the null-space-based behavioral control, IEEE Trans. Control Syst. Technol. 17 (2009) 1173–1182.

[7] J. Ghommem, F. Mnif, G. Poisson, N. Derbel, Nonlinear formation control of a group of underactuated ships, Ocean. 2007 - Eur. (2007).

[8] I. Mas, C. A. Kitts, Dynamic control of mobile multirobot systems: The cluster space formulation, IEEE Access 2 (2014) 558–570.

[9] B. Yu, X. Dong, Z. Shi, Y. Zhong, Formation control for quadrotor swarm systems: Algorithms and experiments, Chinese Control Conf. CCC (2013) 7099–7104.

[10] J. Alonso-Mora, S. Baker, D. Rus, Multi-robot formation control and object transport in dynamic environments via constrained optimization, Int. J. Rob. Res. 36 (2017) 1000–1021.

[11] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, D. Rus, Distributed multi-robot formation control in dynamic environments, Auton. Robots 43 (2018) 1079–1100.

[12] K. Guo, X. Li, L. Xie, Ultra-wideband and Odometry-Based Cooperative Relative Localization With Application to Multi-UAV Formation Control, IEEE Trans. Cybern. (2019) 1–14.

[13] S. K. Pang, Y. H. Li, H. Yi, Joint formation control with obstacle avoidance of towfish and multiple autonomous underwater vehicles based on graph theory and the null-space-based method, Sensors (Switzerland) 19 (2019).

[14] M. Breivik, V. E. Hovstein, T. I. Fossen, Ship formation control: A guided leader-follower approach, volume 17, IFAC, 2008. URL: http://dx.doi.org/10.3182/20080706-5-KR-1001.02706. doi:10.3182/20080706-5-KR-1001.3390.

[15] D. Schoerling, C. Van Kleeck, F. Fahimi, C. R. Koch, A. Ams, P. Löber, Experimental test of a robust formation controller for marine unmanned surface vessels, Auton. Robots 28 (2010) 213–230.

[16] A. Riahifard, S. M. H. Rostami, J. Wang, H. J. Kim, Adaptive leader-follower formation control of under-actuated surface vessels with model uncertainties and input constraints, Appl. Sci. 9 (2019).

[17] Y. Kuwata, J. P. How, Cooperative distributed robust trajectory optimization using receding horizon MILP, IEEE Trans. Control Syst. Technol. 19 (2011) 423–431.

[18] R. R. Negenborn, J. M. Maestre, Distributed model predictive control: An overview and roadmap of future research opportunities, IEEE Control Syst. 34 (2014) 87–97.

[19] J. Alonso-Mora, T. Naegeli, R. Siegwart, P. Beardsley, Collision avoidance for aerial vehicles in multi-agent scenarios, Auton. Robots 39 (2015) 101–121.

[20] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, M. W. Spong, Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments, Int. J. Rob. Res. 27 (2008) 107–126.

[21] S. H. Manoharan, W. Y. Chiu, Consensus Based Formation Control of Automated Guided Vehicles Using Dynamic Destination Approach, 2019 58th Annu. Conf. Soc. Instrum. Control Eng. Japan, SICE 2019 902 (2019) 902–907.

[22] Z. Peng, N. Gu, Y. Zhang, Y. Liu, D. Wang, L. Liu, Path-guided time-varying formation control with collision avoidance and connectivity preservation of under-actuated autonomous surface vehicles subject to unknown input gains, Ocean Eng. 191 (2019).

[23] A. Perizzato, M. Farina, R. Scattolini, Formation control and collision avoidance of unicycle robots with distributed predictive

control, IFAC-PapersOnLine 23 (2015) 260–265.
[24] H. Zheng, J. Wu, W. Wu, R. R. Negenborn, Cooperative distributed predictive control for collision-free vehicle platoons, IET Intell. Transp. Syst. 13 (2019) 816–824.
[25] J. Wang, S. Gong, S. Peeta, L. Lu, A real-time deployable model predictive control-based cooperative platooning approach for connected and autonomous vehicles Highlights :, Elsevier 128 (2019) 271–301.
[26] M. Abdelaal, M. Fränzle, A. Hahn, NMPC-based Trajectory Tracking and Collison Avoidance of Underactuated Vessels with Elliptical Ship Domain, IFAC-PapersOnLine 49 (2016) 22–27.
[27] L. Chen, Y. Huang, H. Zheng, H. Hopman, R. Negenborn, Cooperative Multi-Vessel Systems in Urban Waterway Networks, IEEE Trans. Intell. Transp. Syst. (2019) 1–14.
[28] T. I. Fossen, Handbook of Marine Craft Hydrodynamics and Motion Control, 2011. URL: http://onlinelibrary.wiley.com/book/10.1002/9781119994138. doi:10.1002/9781119994138.
[29] W. Wang, L. A. Mateos, S. Park, P. Leoni, B. Gheneti, F. Duarte, C. Ratti, D. Rus, Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle (2018).
[30] H. Zheng, R. R. Negenborn, G. Lodewijks, Predictive path following with arrival time awareness for waterborne AGVs, Transp. Res. Part C Emerg. Technol. 70 (2016) 214–237.
[31] A. Haseltalab, R. R. Negenborn, Model predictive maneuvering control and energy management for all-electric autonomous ships, Appl. Energy 251 (2019).
[32] A. Veksler, T. A. Johansen, F. Borrelli, B. Realfsen, Dynamic Positioning With Model Predictive Control, IEEE Trans. Control Syst. Technol. 24 (2016) 1340–1353.
[33] Stanford Artificial IntelligenceLaboratory, Robotic Operating System, 2018. URL: https://www.ros.org.
[34] B. Houska, H. Ferreau, M. Diehl, ACADO Toolkit - An Open Source Framework for Automatic Control and Dynamic Optimization, Optim. Control Appl. Methods 32 (2011) 298–312.
[35] A. Haseltalab, V. Garofano, M. Van Pampus, R. Negenborn, Model Predictive Trajectory Tracking Control and Thrust Allocation for Autonomous Vessels, Submitted (2019).

Appendix A. PI control

A standard MIMO PI controller with bounded output is designed. The tuning parameters are given in Appendix C. The output of the PI controller is a vector with reference forces and moment in the body-fixed frame $\tau_d(k)$. Using this vector, the control inputs are determined using the following thrust allocation algorithm.

Roboat has an over-actuated thrust allocation, that is addressed with an optimization function that minimizes the amount of force and the amount of change of force per thruster. The algorithm used is the following:

$$J = \min_{\Delta\mathbf{u}(k)} \Big(\mathbf{u}(k-1)+\Delta\mathbf{u}(k)\Big)^{\mathrm{T}} \mathbf{P}_{\mathrm{TA}} \Big(\mathbf{u}(k-1)+\Delta\mathbf{u}(k)\Big) + \Delta\mathbf{u}(k)^{\mathrm{T}} \mathbf{Q}_{\mathrm{TA}} \Delta\mathbf{u}(k) \quad \text{(A.1)}$$

subject to

$$\tau_d(k) = \mathbf{B}\Big(\mathbf{u}(k-1) + \Delta\mathbf{u}(k)\Big) \quad \text{(A.2)}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(k-1) + \Delta\mathbf{u}(k) \leq \mathbf{u}_{\max} \quad \text{(A.3)}$$

$$\Delta\mathbf{u}_{\min} \leq \Delta\mathbf{u}(k) \leq \Delta\mathbf{u}_{\max} \quad \text{(A.4)}$$

where $\Delta\mathbf{u}(k) \in \mathbb{R}^{4\times1}$ is the change in force per actuator; $\mathbf{u}(k-1) \in \mathbb{R}^{4\times1}$ is the force per actuator; $\mathbf{P}_{\mathrm{TA}} \in \mathbb{R}^{4\times4}_{>0}$ and $\mathbf{Q}_{\mathrm{TA}} \in \mathbb{R}^{4\times4}_{>0}$ are positive definite penalty matrices; $\tau_d \in \mathbb{R}^{3\times1}$ is the vector with reference internal forces and moments; $\mathbf{B} \in \mathbb{R}^{3\times4}$ is the thrust allocation matrix; $\mathbf{u}_{\min} \in \mathbb{R}^{4\times1}$, $\mathbf{u}_{\max} \in \mathbb{R}^{4\times1}$, $\Delta\mathbf{u}_{\min} \in \mathbb{R}^{4\times1}$ and $\Delta\mathbf{u}_{\max} \in \mathbb{R}^{4\times1}$ are the limits of the actuators. The values of the parameters used, can be found in Appendix C. This method is a simplification of the model predictive thrust allocation algorithm that has been designed in [35].

Appendix B. Determination of the KPIs

To assess the various methods, the following KPIs are used:

1. Root mean square error (RMSE) of the reference trajectory relative to the global trajectory (guidance system):

$$\sqrt{\frac{\sum_{k=1}^{N}\Big(\big(\|\mathbf{p}_d(k) - \tilde{\mathbf{p}}(k)\|\big)^2\Big)}{N}} \quad \text{(B.1)}$$

2. RMSE of the measured position relative to the reference trajectory (control system):

$$\sqrt{\frac{\sum_{k=1}^{N}\Big(\big(\|\hat{\mathbf{p}}(k) - \mathbf{p}_d(k)\|\big)^2\Big)}{N}} \quad \text{(B.2)}$$

3. RMSE of the measured heading relative to reference heading (control system):

$$\sqrt{\frac{\sum_{k=1}^{N}\Big(\big(\left\|\hat{\psi}(k) - \psi_d(k)\right\|\big)^2\Big)}{N}} \quad \text{(B.3)}$$

4. Square root of the total sum of forces applied by each thruster squared (control system):

$$\sqrt{\sum_{k=1}^{N}\Big(\big(\|\mathbf{u}(k)\|\big)^2\Big)} \quad \text{(B.4)}$$

5. Square root of the total sum of the differences in successive control inputs of each thruster over time squared (control system):

$$\sqrt{\sum_{k=1}^{N-1}\Big(\big(\|\mathbf{u}(k+1) - \mathbf{u}(k)\|\big)^2\Big)} \quad \text{(B.5)}$$

Appendix C. Overview of parameters used in simulations and real experiments

Appendix D. Characteristics and properties of Roboat

Appendix E. Results of Test 6 in simulation and real experiments

Table C.2: An overview of the parameters used during all tests

Category	Parameter	Value
Control system settings	t_s	0.5 [s]
	$\tau_{\min}$[a]	$\begin{bmatrix}-4\\-4\\-2\end{bmatrix}$ [N] [N] [Nm]
	$\tau_{\max}$[a]	$\begin{bmatrix}4\\4\\2\end{bmatrix}$ [N] [N] [Nm]
Thrust allocation settings	$\mathbf{P}_{TA}$[a]	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&10&0\\0&0&0&10\end{bmatrix}$
	$\mathbf{Q}_{TA}$[a]	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$
	$\mathbf{u}_{\min}$	$\begin{bmatrix}-2&-2&-2&-2\end{bmatrix}^T$ [N]
	$\mathbf{u}_{\max}$	$\begin{bmatrix}2&2&2&2\end{bmatrix}^T$ [N]
	$\Delta\mathbf{u}_{\min}$	$\begin{bmatrix}-1&-1&-1&-1\end{bmatrix}^T$ [Ns^{-1}]
	$\Delta\mathbf{u}_{\max}$	$\begin{bmatrix}1&1&1&1\end{bmatrix}^T$ [Ns^{-1}]
Guidance system settings	d_{safe}	1.5 [m]
	$v_{\max}$	1.0 [ms^{-1}]
	t_{12}	5 [s]
	t_{13}	5 [s]
	y_{12}	−2 [m]
	y_{12}	2 [m]
Input	Waypoints	$(-5\ \ -5)$ $(-15\ \ -5)$ $(-15\ \ -25)$ $(-25\ \ -25)$ $(-25\ \ -5)$ $(-15\ \ -5)$ $(-15\ \ -25)$ $(-5\ \ -25)$ $(-5\ \ -5)$
	v_d	0.2 [ms^{-1}]
	T	300 [s]

[a] Does not apply to NMPC method

Table C.3: An overview of the tuning parameters for PID control

Parameter	Simulation	Real experiments
K_p	$\begin{bmatrix}3.5&0&0\\0&3.0&0\\0&0&2.0\end{bmatrix}$	$\begin{bmatrix}0.96&0&0\\0&0.96&0\\0&0&0.56\end{bmatrix}$
K_i	$\begin{bmatrix}0.001&0&0\\0&0.001&0\\0&0&0.001\end{bmatrix}$	$\begin{bmatrix}0.017&0&0\\0&0.017&0\\0&0&0.017\end{bmatrix}$
K_d	$\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}$

Table C.4: An overview of the tuning parameters for FL MPC

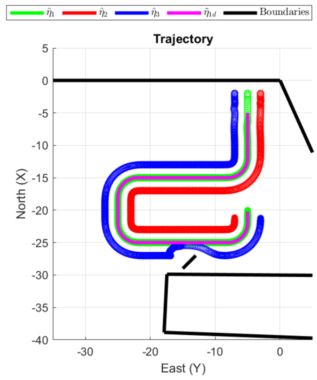
Parameter	Simulation	Real experiments
$\mathbf{Q}_{FL}$	$\begin{bmatrix}100&0&0\\0&100&0\\0&0&100\end{bmatrix}$	$\begin{bmatrix}10&0&0\\0&10&0\\0&0&10\end{bmatrix}$
β	3	4

Table C.5: An overview of the tuning parameters for NMPC

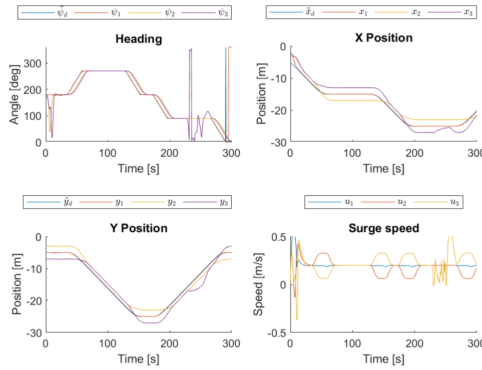
Parameter	Simulation & real experiments
Q_{NMPC}	$\begin{bmatrix}20&0&0&0&0&0\\0&20&0&0&0&0\\0&0&5&0&0&0\\0&0&0&0.001&0&0\\0&0&0&0&0.001&0\\0&0&0&0&0&0.001\end{bmatrix}$
R_{NMPC}	$\begin{bmatrix}5&0&0&0\\0&5&0&0\\0&0&5&0\\0&0&0&5\end{bmatrix}$

Table D.6: An overview of the properties of the 1:4 scale Roboat

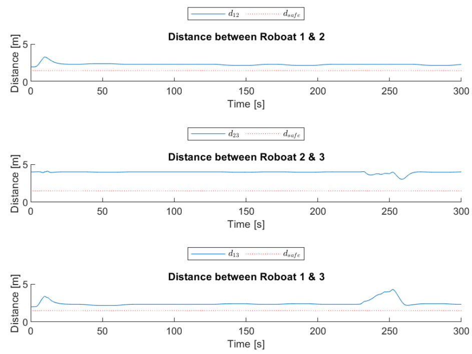
Sector	**Property**	**Value / type**
Physical properties	Dimensions	0.90 by 0.45 by 0.15 [m]
	Total mass	9.2 [kg]
Sensors	RTK GPS	Emlid Reach RS+
	IMU	MicroStrain 3DM-GX5-25
Computer & communication	Wi-Fi antenna	Alfa Network AWUS036NH
	Mini pc	Intel Dawson Canyon NUC7i7DNH
	Data processor	STM32F103
Actuators & power	Thrusters	4 x T100 Thruster
	Battery	11.1 V rechargeable Li-Po
	Power converter	9-36V to 12V & 19V
	Operation time	2 [h]



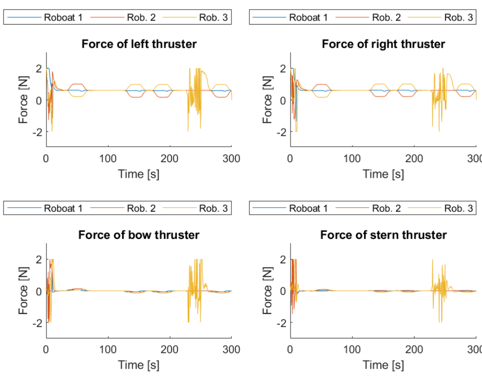
(a) The sailed trajectory



(b) Position over time and surge speed

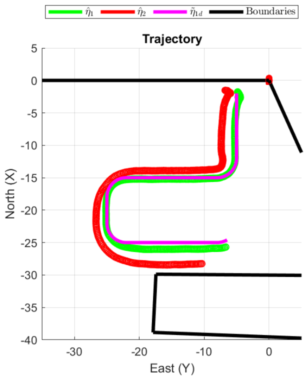


(c) Distances to other vessels

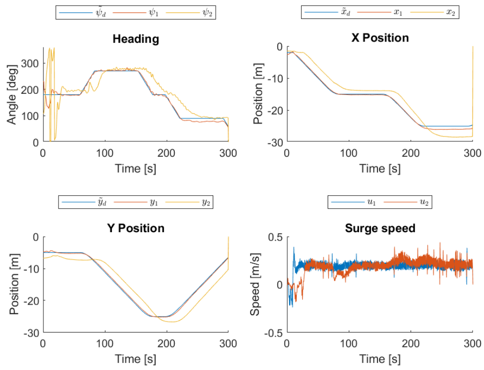


(d) Control input over time

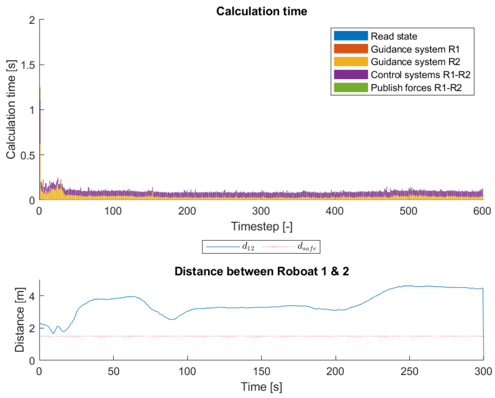
Figure E.7: The results of simulation test 6: NMPC & half guidance system



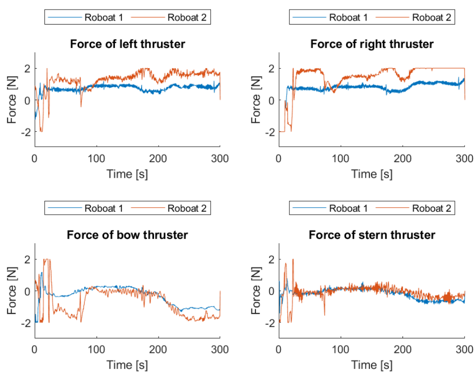
(a) The sailed trajectory



(b) Position over time and surge speed



(c) Distances to other vessels



(d) Control input over time

Figure E.8: The results of real experiment test 6: NMPC & half guidance system leader

B

Information waste collection provided by municipality of Amsterdam

On request the manager of industrial waste collection of the municipality of Amsterdam provided the following information about the collection of household waste in *Amsterdam Centrum*.

Table B.1: Information provided by the manager of industrial waste collection of the municipality of Amsterdam

Item	**Quantity**	**Unit**
Amount of waste trucks dedicated for *Amsterdam Centrum*	14	[trucks]
Average amount of routes for waste trucks per working day for *Amsterdam Centrum*	9	[routes / day]
Average amount of hours per route	8	[hours / route]
Amount of working days per week	6	[working days / week]
Amount of employees per waste truck	3	[employees / truck]
Purchase cost per truck	ca. 300,000	[EUR / truck]
Economic life span of a waste truck	7	[years]
Weight of an empty waste truck	20	[MT]
Weight of a full waste truck	28	[MT]
Capacity of a waste truck	10	$[m^3]$
Density of household waste after pressing	700	$[kg/m^3]//$

C

Cost estimation of waste collection in Amsterdam Centrum

The total costs of the waste collection in *Amsterdam Centrum* are not clear. However an estimation has been made to give an indication of the total costs in the current situation and in a future situation where Roboats collect the waste.

In Table C.1 the cost estimation of the current situation is shown.

Table C.1: Rough estimation on current costs of household waste collection in *Amsterdam Centrum*

Cost center	**Costs**	**Unit**	**Amount**	**Cost per year**
Purchase Costs	300,000 [a]	EUR / (truck * 7 years)	14 trucks [a]	EUR 600,000
Maintenance Costs	5000 [b]	EUR / (truck * year)	14 trucks [a]	EUR 70,000
Personnel Salary	32 [c]	EUR / hour	67,400 hours / year [a,d]	EUR 2,157,000
Diesel costs	1.169 [e]	EUR / L	132,000 L / year [a,f,g]	EUR 154,000
Total Costs				EUR 2,981,000

[a] Provided by the municipality of Amsterdam, see Appendix B
[b] Assumption made by the author
[c] According to (Avalex (2019))
[d] 8 [hours / route] * 9 [routes / day] * 3 [employees / truck] * 6 [days / week] * 52 [weeks / year] = 67,392 [hours / year], see Appendix B
[e] According to (Brandstof-zoeker.nl (2020))
[f] A waste truck in Amsterdam consumes on average 47 liter diesel per day according to (Fleet&Mobility (2018))
[g] $47\left[\frac{\text{L}}{\text{day} * \text{route}}\right] * 9\left[\frac{\text{routes}}{\text{day}}\right] * 6\left[\frac{\text{days}}{\text{week}}\right] * 52\left[\frac{\text{weeks}}{\text{year}}\right] = 131,976\left[\frac{\text{L}}{\text{year}}\right]$

The minimum amount of dumpsters necessary is calculated by (Daub (2019)) including refernces. The researchers determined that 283 collection locations will be sufficient to collect the waste in *Amsterdam Centrum* using scenario 1: Residents can dump their waste twice a week on a specified day, either between 07:30 to 15:30 or after 17.00. Using six collection days per week and two collection timeslots per day, the researchers created six collection zones that will be served twice a week each. They determined that 48 floating dumpsters with a capacity of 1 MT each were enough to complete the waste collection during every timeslot. Since on average every week 503 MT of waste is being collected, as explained in Section 1.1.1, the average waste collection per timeslot equals:

$$\frac{503\left[\frac{\text{MT}}{\text{week}}\right]}{6\left[\frac{\text{days}}{\text{week}}\right]*2\left[\frac{\text{timeslots}}{\text{day}}\right]} = 42\left[\frac{\text{MT}}{\text{timeslot}}\right] \tag{C.1}$$

The maximum amount of waste that can be collected during one timeslot according to (Daub (2019)) equals 48 MT. Therefore it is doubted that the number of floating dumpsters calculated by (Daub (2019)) is sufficient to transport all the waste, also during peak moments. On the other hand, it is also suspected that the capacity of a dumpster with dimensions of 6 by 2 meters (Daub (2019)), can be higher than 1 MT. For the first rough cost estimation, it is assumed that 48 floating dumpsters will be sufficient.

In (Daub (2019)) the minimum amount of Roboats - that function as tugboats - is determined at 4. According to this master thesis, the operating time per Roboat per zone is less than 8 hours when the Roboat can move up to 3 floating dumpsters. However in their calculation only the collection of full dumpsters is taken into account, the placement of empty dumpsters is disregarded. It is assumed that for placing empty dumpsters on collection locations on time, another 4 Roboats are necessary. Due to the time necessary for the process of emptying the dumpsters and placing new ones simultaneously as the collection of dumpsters, it is also assumed that the double amount of calculated dumpsters will be necessary: 96 dumpsters.

Although at the moment it is impossible to know what the costs of a Roboat would be, it would be very helpful to have an estimation. For the moment it is estimated that the purchase cost of one Roboat would be circa EUR 125,000 with an economical lifespan of 5 years. An explanation on this cost estimation can be found in Table C.3. This estimation is based on conversations with the project manager of Roboat and serves only as an indication. Furthermore the costs of a dumpster with dimensions of 6 by 2 meters and a capacity of minimal 1 MT are estimated on EUR 10,000, with an economical lifespan of 10 years. With that information, a very rough cost estimation of waste transport using Roboats under scenario 1 is presented in Table C.2. In this cost estimation it is assumed that:

1. Roboats are able to operate 16 hours per day;
2. Two backup Roboats are sufficient to cover maintenance work and extra capacity during peak moments;
3. A total of 96 dumpsters will be enough to transport all the household waste and no backup dumpsters are necessary;
4. No extra personnel is necessary;
5. The solution to transport the waste from the dumpster or Roboat to the storage unit does not cost extra money or personnel;
6. The solution to transport the waste from the storage unit to the incinerator of *AEB* does not cost extra money or personnel.

Other assumptions that have been made during the estimation of the costs are mentioned in the table. It is assumed that no extra personnel is necessary (Assumption 4) since managers and planners that used to work with waste trucks, can be deployed to monitor the Roboats. In the case that this is not sufficient and for example two operators are necessary every moment of the day, this could lead to extra costs: If the operators cost EUR 50 per hour per operator, working 24 hours per day for 6 days per week, this would cost an additional EUR 375,000 per year. Furthermore the testing and starting phase, in which probably more personnel would be necessary, is not included in this cost estimation.

Table C.2: Rough estimation on costs of household waste collection in *Amsterdam Centrum* using autonomous Roboats

Cost center	Costs	Unit	Amount	Cost per year
Purchase Costs dumpsters	10,000 [a]	EUR / (dumpster * 10 years)	96 dumpsters [b]	EUR 96,000
Purchase Costs Roboats	125,000 [a]	EUR / (Roboat * 5 years)	10 Roboats [c]	EUR 250,000
Maintenance Costs	10000 [a]	EUR / (Roboat * year)	10 Roboats [c]	EUR 100,000
Personnel Salary	0 [a]	EUR / hour	0 hours / year [a]	EUR 0
Electricity costs	0.089 [d]	EUR / kWh	900,000 kWh / year [e,f]	EUR 80,000
Total Costs				EUR 526,000

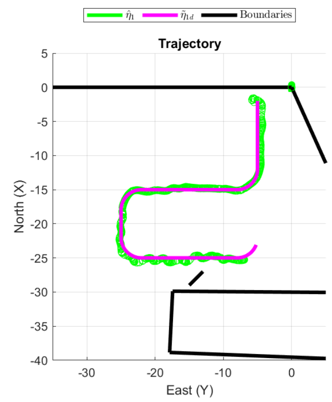
[a] Assumption made by the author
[b] According to (Zhang et al. (2020))
[c] According to (Daub (2019))
[d] According to the online cost calculator of energy company *Essent* via `www.essent.nl` using the estimated power consumption
[e] It is assumed that on average 30 hp, or 22 kW is consumed
[f] $8[\text{Roboats}] * 6\left[\frac{\text{days}}{\text{week}}\right] * 52\left[\frac{\text{weeks}}{\text{year}}\right] * 16\left[\frac{\text{hours}}{\text{day} * \text{Roboat}}\right] * 22[\text{kW}] = 878,592\left[\frac{\text{kWh}}{\text{year}}\right]$

Table C.3: Rough estimation on production costs of one Roboat

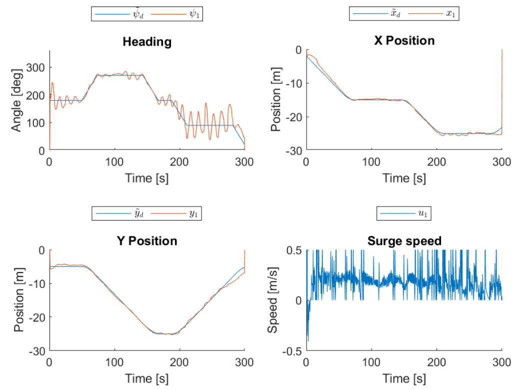
Cost center	Cost
Hull	EUR 30,000
Lidars (2x)	EUR 20,000
Computer + other sensors	EUR 5,000
Battery pack	EUR 20,000
Thrusters	EUR 20,000
Small hardware / electronics	EUR 5,000
Development costs	EUR 25,000
Total Costs	EUR 125,000

D
Parameter tuning

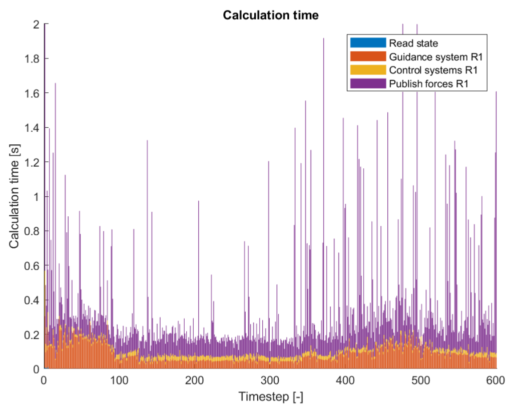
In this appendix, the results of the experiments with the final tuning parameters are given. In each experiment, one Roboat had to sail a predetermined trajectory. During the tuning process, some bugs in the implemented structure have been discovered and fixed afterwards. For example the speed was determined incorrectly and the process of sending the control input to the ROS network from the Matlab laptop took too much time due to an error.



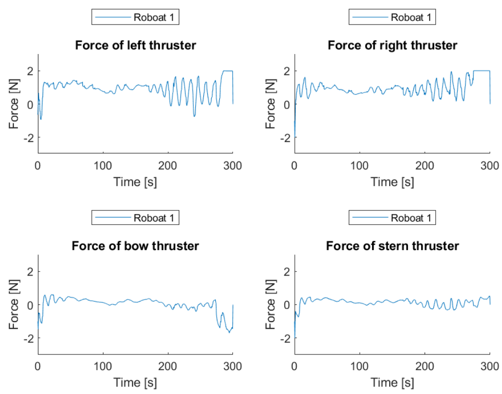
(a) Sailed trajectory



(b) Position over time and surge speed



(c) Distances to other vessels



(d) Control input over time

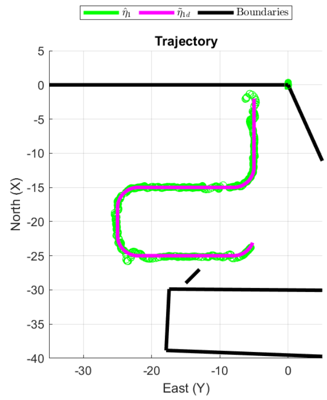
Figure D.1: Results of the experiment with final tuning parameters for physical experiments with PID control system

Table D.1: KPIs during the tuning process for the physical experiments with PID control system

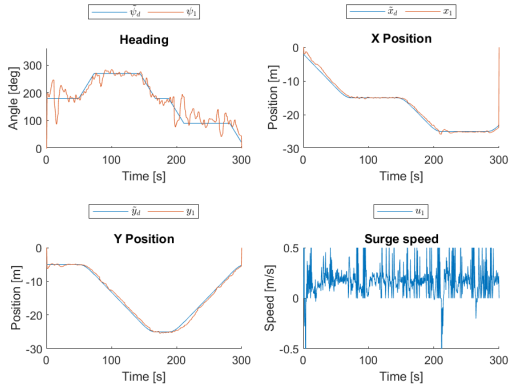
KPI	Value (Roboat 1)
KPI 1: RMSE trajectory guidance system	0
KPI 2: RMSE trajectory control system	1.14
KPI 3: RMSE heading control system	1.64
KPI 4: Applied forces by thrusters	39.4
KPI 5: Change of control input	6.2

Table D.2: KPIs during the tuning process for the physical experiments with FLMPC control system

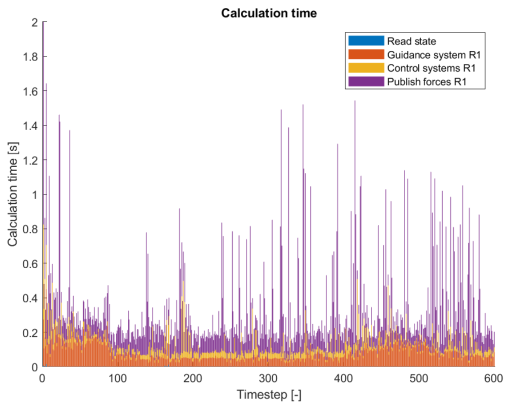
KPI	Value (Roboat 1)
KPI 1: RMSE trajectory guidance system	0
KPI 2: RMSE trajectory control system	1.07
KPI 3: RMSE heading control system	0.70
KPI 4: Applied forces by thrusters	57.2
KPI 5: Change of control input	33.8



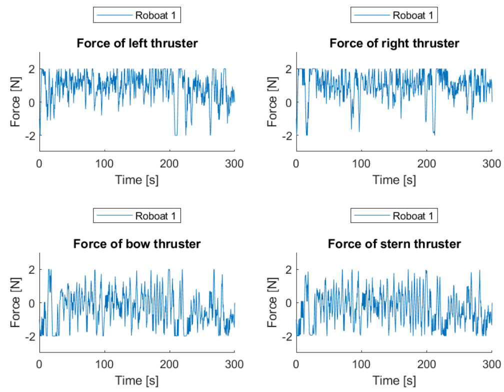
(a) Sailed trajectory



(b) Position over time and surge speed



(c) Distances to other vessels

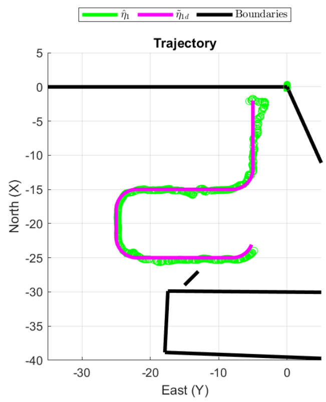


(d) Control input over time

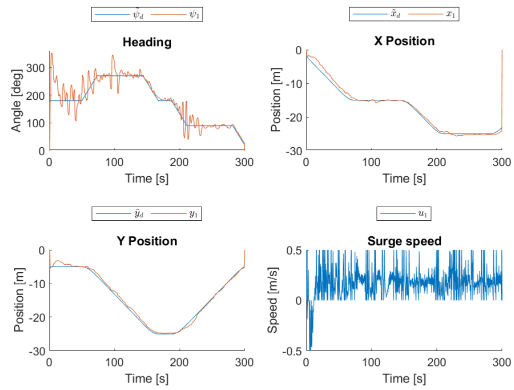
Figure D.2: Results of the experiment with final tuning parameters for physical experiments with FLMPC control system

Table D.3: KPIs during the tuning process for the physical experiments with NMPC control system

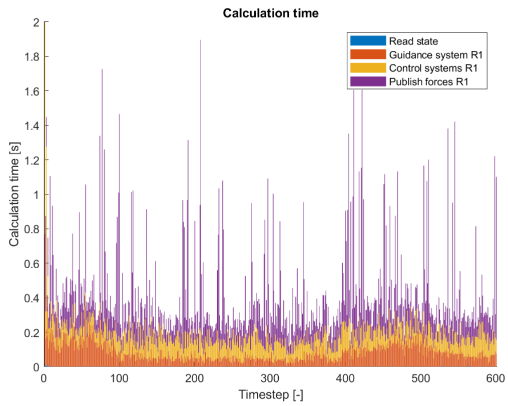
KPI	Value (Roboat 1)
KPI 1: RMSE trajectory guidance system	0
KPI 2: RMSE trajectory control system	1.08
KPI 3: RMSE heading control system	0.76
KPI 4: Applied forces by thrusters	52.2
KPI 5: Change of control input	36.8



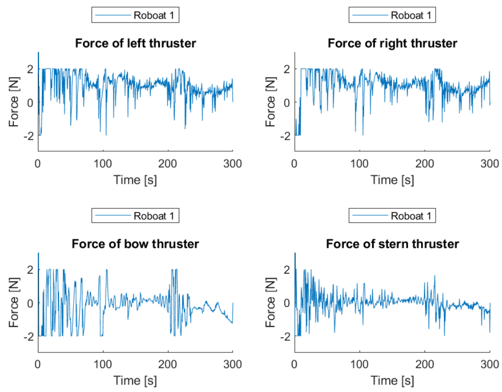
(a) Sailed trajectory



(b) Position over time and surge speed



(c) Distances to other vessels



(d) Control input over time

Figure D.3: Results of the experiment with final tuning parameters for physical experiments with NMPC control system

E

Sensor test

In this appendix, the results of the experiments to assess the RTK and IMU sensors are shown.

Table E.1: Results of the ten experiments to assess the RTK GPS and IMU sensors

Test	X: Avg error[a]	X: MAD mean[b]	X: MAD abs[c]	ψ: Avg error[a]	ψ: MAD mean[b]	ψ: MAD abs[c]	v_y[d]	v_y: Avg error[a,d]	v_y: % within 0.3 m/s[e]
1	-3.83 [m]	0.12 [m]	3.83 [m]	6.9 [deg]	2.6 [deg]	7.3 [deg]	-1.46 [m/s]	0.03 [m/s]	45 [%]
2	-1.60 [m]	0.04 [m]	1.60 [m]	6.9 [deg]	2.9 [deg]	7.7 [deg]	-1.36 [m/s]	0.03 [m/s]	56 [%]
3	-1.13 [m]	0.06 [m]	1.13 [m]	7.5 [deg]	1.6 [deg]	7.5 [deg]	-1.46 [m/s]	0.05 [m/s]	52 [%]
4	-1.09 [m]	0.03 [m]	1.09 [m]	7.3 [deg]	2.2 [deg]	7.3 [deg]	-1.20 [m/s]	0.03 [m/s]	77 [%]
5	-0.71 [m]	0.05 [m]	0.71 [m]	4.6 [deg]	2.9 [deg]	6.0 [deg]	-1.15 [m/s]	0.04 [m/s]	74 [%]
6	-0.45 [m]	0.04 [m]	0.45 [m]	2.1 [deg]	1.0 [deg]	2.4 [deg]	-1.30 [m/s]	0.04 [m/s]	57 [%]
7	-0.53 [m]	0.04 [m]	0.53 [m]	1.1 [deg]	0.9 [deg]	1.4 [deg]	-1.50 [m/s]	0.06 [m/s]	44 [%]
8	-0.69 [m]	0.04 [m]	0.69 [m]	0.3 [deg]	0.9 [deg]	0.9 [deg]	-1.46 [m/s]	0.05 [m/s]	45 [%]
9	-0.62 [m]	0.03 [m]	0.62 [m]	1.2 [deg]	1.0 [deg]	1.6 [deg]	-1.11 [m/s]	0.03 [m/s]	69 [%]
10	-0.49 [m]	0.03 [m]	0.49 [m]	0.7 [deg]	0.8 [deg]	1.0 [deg]	-1.18 [m/s]	0.03 [m/s]	77 [%]

[a] Avg: average
[b] MAD mean: mean absolute deviation around mean
[c] MAD abs: mean absolute deviation around true value
[d] $\mathbf{v_y}$: speed in global y-direction
[e] The percentage of timesteps in which the measured speed is within 0.3 m/s of the true value