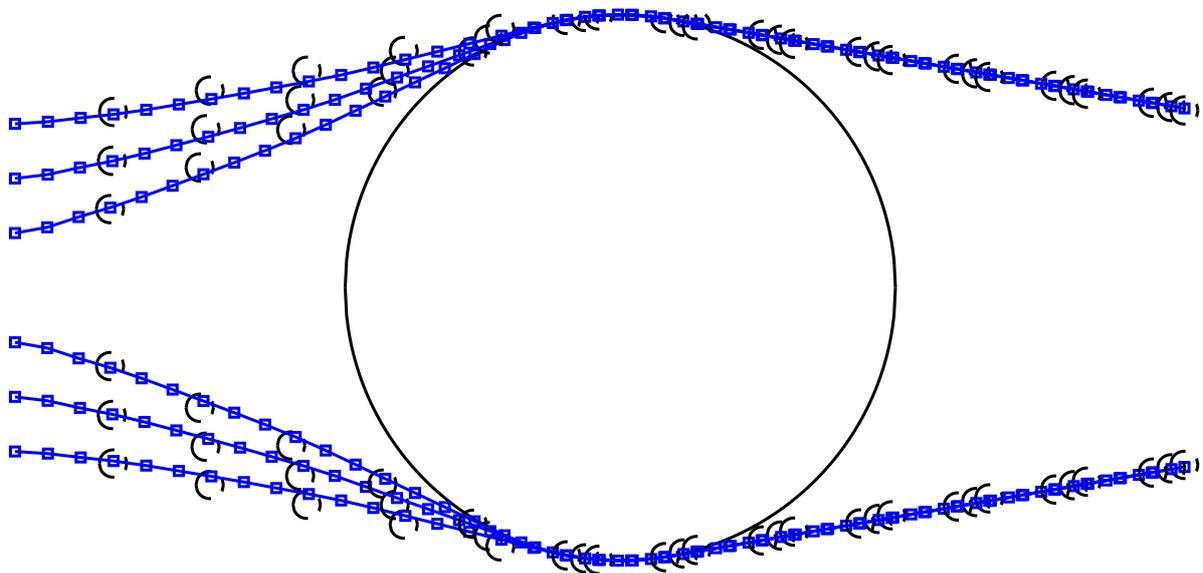


An Exploration of Integrated Bi-level Local and Global Model Predictive Control for Air Traffic Control Applications

D. Tol



An Exploration of Integrated Bi-level Local and Global Model Predictive Control for Air Traffic Control Applications

by

D. Tol

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday June 29th, 2020 at 10:00 AM.

Student number:	4290275	
Project duration:	August 2019 – June 2020	
Thesis committee:	Dr. A. Jamshidnejad,	TU Delft, daily supervisor
	Prof. dr. ir. J. M. Hoekstra,	TU Delft, supervisor
	Dr. ir. A.J.J. van den Boom	TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

Dear reader,

This MSc. thesis project with the title, "An Exploration of Integrated Bi-level Local and Global Model Predictive Control for Air Traffic Control Applications" concludes a challenging, enjoyable and memorable time as a student at the Faculty of Aerospace Engineering at the Delft University of Technology.

I would like to express my gratitude to both my supervisors Anahita Jamshidnejad and Jacco Hoekstra for their guidance and support throughout this challenging project. Their time and valuable feedback are greatly appreciated and helped to make this research project possible. The expertise and enthusiasm they expressed during our meetings have been a huge source of motivation. It has been a truly enjoyable experience, which helped me grow as an engineer and as a person.

I am thankful to my daily supervisor Anahita for providing me with the conceptual framework of this subject and giving me the opportunity to work on this exciting topic.

I wish to thank my parents and my sister for their love and support during my studies. Finally, I want to thank all my friends who have contributed to creating a truly wonderful and unforgettable time in Delft.

Thank you.

D. Tol
Delft, June 2020

Thesis Structure

1. **Scientific Paper**, containing the most important findings and relevant conclusions of this thesis in the form of a scientific paper.
2. **Thesis Appendices**, additional results and relevant information supporting the results in the thesis.
3. **Preliminary Report & Literature Study**, already graded report containing background information, literature study and preliminary results. The list of symbols, tables and figures correspond to this part of the report. Added for completeness and reference.

Contents

Thesis Structure	v
List of Figures	ix
List of Tables	xiii
List of Symbols (Preliminary Report)	xvi
List of Abbreviations	xvii
I Scientific Paper	1
II Thesis Appendices	31
III Preliminary Report & Literature Study Already Graded	45
1 Introduction	47
1.1 Project Scope	47
1.2 Research Questions	47
1.3 Research Objectives	48
1.4 Preliminary Report Structure	49
2 Model Predictive Control Working Principles	51
2.1 Working Principles of Model Predictive Control	51
2.1.1 Prediction Model	52
2.1.2 Cost Function	52
2.1.3 Constraints	52
2.1.4 Advantage and Disadvantages of Model Predictive Control	53
2.2 Characteristics of Air Traffic Control Optimization	53
2.3 Aircraft Model	54
3 Decentralization in Air Traffic Management: Motivation and Concepts	55
3.1 Centralized, Decentralized and Distributed Air Traffic Management	55
3.2 Conflict Detection & Avoidance Concepts	57
3.3 4D-Trajectory Methods	58
3.4 Optimization Based Decentralized Control in Air Traffic Management	58
4 Decentralized and Multi-Level MPC Architectures	61
4.1 Multi-level Control and Control System Architecture	61
4.2 Setpoint Optimization	61
4.3 Decentralized and Distributed Control	62
4.4 Applications of Multi-Level Controllers	64
5 Bi-Level Model Predictive Controller Conceptual Design	67
5.1 Bi-level Controller Working Principles	67
5.2 Concept Motivation From Literature	68
5.3 Practical Consideration and Link to Reality	69
5.4 Controller Integration and Conceptual Development	69
5.4.1 Concept 1: Heading and Velocity Constraints	69
5.4.2 Concept 2: Longitudinal and Velocity Constraints	70
5.4.3 Concept 3: Combination of Concept 1 and Concept 2	70
5.4.4 Concept 4: Maximum Position Constraint Relative to Setpoints	71
5.4.5 Real-Time Simulation Consideration	71

5.4.6	Extension of the Decentralized Prediction Horizon	72
5.4.7	Determination of the Constraint Boundaries.	73
5.5	Bi-Level Model Predictive Control Formulation	74
5.5.1	Centralized and Decentralized Cost Functions.	75
5.5.2	Centralized and Decentralized Constraints	75
5.5.3	Centralized Separation Constraint	75
5.5.4	Decentralized Maximum Position From Setpoints Constraints.	76
6	Thesis Project Timeline and Controller Development Plan	77
6.1	Project Planning and Gantt Chart	77
6.2	Development Phase I - Preliminary Development Stage	78
6.3	Development Phase II - Controller Development and Bi-level Integration	79
6.4	Development Phase III - Bi-level Controller Implementation in BlueSky	80
7	Experiment Design	81
7.1	Experiment Phase I: Small Scale Controller Testing	81
7.2	Experiment Phase II: Conflict Avoidance	82
7.3	Experiment Phase III: Multiple Aircraft Simulation	82
8	Preliminary Experiment Results	83
8.1	Experiment 1: Decentralized Static Obstacle Avoidance	83
8.1.1	Experiment 1: Experiment Setup and Performance Metrics	83
8.1.2	Experiment 1: Simulation Results	84
8.2	Experiment 2: Number of Setpoint Determination	88
8.2.1	Experiment 2: Setpoint Testing Experiment Setup	88
8.2.2	Experiment 2: Simulation Results	89
8.2.3	Experiment 2: Setpoints Boundary Size Influence	94
8.2.4	Experiment 2: Increase in Experiment Iterations.	96
8.3	Experiment 3: Larger Forbidden Area	98
8.4	Experiment 4: Single Setpoint Prediction Window Influence	99
8.4.1	Experiment 4: Final Experiment Setup	99
8.4.2	Experiment 4: Final Experiment Setup	100
9	Conclusion	103
A	Appendix A: Gantt Chart	105
B	Appendix B: Initial Simulation Results	107
C	Appendix C: Boeing 747 Model Parameters	113
	Bibliography	115

List of Figures

2.1	Graphical representation of the moving horizon principle.	52
2.2	Zero-order hold approach to piece-wise continuous control inputs.	54
3.1	Decentralized air traffic control structure compared to a centralized structure.	56
3.2	Minimum safety margins around en-route aircraft.	57
4.1	A hierarchical control structure.	62
4.2	Decomposition of different dynamic rates within the system.	65
4.3	Different update rates at different levels of the controller.	65
5.1	Difference in optimization control sample time and the resulting state trajectories.	67
5.2	Centralized conflict-free trajectories determination using setpoint while avoiding forbidden area.	68
5.3	Layered structure in air traffic control.	69
5.4	Concept 1, heading and velocity constraints.	70
5.5	Concept 2, velocity and longitudinal constraint with axis transformation	70
5.6	Concept 3, combined longitudinal and heading constraints with velocity constraints.	71
5.7	Concept 4, setpoints to area constraints conversion.	71
5.8	Schematic representation of the update rate of the decentralized and centralized controllers.	72
5.9	Propagation of an aircraft over time with intermediate constraint updates by the centralized controller.	72
5.10	Increase in decentralized prediction horizon to include multiple setpoints.	73
5.11	Decrease in constraint sizes when distance to destination decreases.	73
5.12	Bi-level controller control diagram.	74
5.13	Possible conflicts even though separation constraints on setpoints are obeyed when the setpoints are to far apart or the separation distance is to small.	76
5.14	Bringing setpoints closer together solves this problem, by making this solution infeasible.	76
6.1	Global thesis timeline and project tasks.	77
6.2	Main development milestones.	78
6.3	Development flowchart for Phase I.	78
6.4	Development flowchart for Phase II.	79
6.5	Development flowchart for Phase III.	80
7.1	Experiment phase I, Aircraft with random initial conditions navigate through an area of random obstacles towards a randomly placed waypoint.	81
7.2	Experiment phase II, bi-level separation constraint testing using two aircraft.	82
7.3	Experiment phase III, multi-aircraft testing in BlueSky.	82
8.1	Derivation of Equation 8.2.	84
8.2	Aircraft trajectory with static obstacle, with $H_p = 5$ and control sample time = 5 seconds.	85
8.3	Aircraft trajectory with static obstacle, with $H_p = 10$ and control sample time = 5 seconds.	85
8.4	Aircraft trajectory with static obstacle, with $H_p = 15$ and control sample time = 5 seconds.	85
8.5	Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 5 seconds.	85
8.6	Close up of the aircraft trajectory around static obstacle, with $H_p = 5$ and control sample time = 5 seconds.	85
8.7	Close up of the aircraft trajectory around static obstacle, with $H_p = 20$ and control sample time = 10 seconds.	85
8.8	Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 5 seconds.	86
8.9	Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 5 seconds.	86
8.10	Aircraft velocity with static obstacle, with $H_p = 5$ and control sample time = 5 seconds.	86

8.11 Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 5 seconds.	86
8.12 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 5$ and control sample time = 5 seconds.	87
8.13 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 10$ and control sample time = 5 seconds.	87
8.14 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 15$ and control sample time = 5 seconds.	87
8.15 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 20$ and control sample time = 5 seconds.	87
8.16 Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 15 seconds.	88
8.17 Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 20 seconds.	88
8.18 Aircraft velocity with static obstacle, with $H_p = 20$ and control sample time = 15 seconds.	88
8.19 Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 20 seconds.	88
8.20 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 2.	90
8.21 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 8, initial condition iteration 1.	90
8.22 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 6.	90
8.23 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 8, initial condition iteration 10.	90
8.24 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 5.	91
8.25 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 7, initial condition iteration 5.	91
8.26 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 1.	91
8.27 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 10.	91
8.28 Forbidden area violation count with a setpoint boundary of 1500m.	92
8.29 Fuel consumption per simulation iteration for 8 experiment sets.	92
8.30 Remaining distance to final destination per simulation iteration for 8 experiment sets.	93
8.31 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 5.	93
8.32 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 2, initial condition iteration 4.	93
8.33 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 4, initial condition iteration 8.	94
8.34 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 4, initial condition iteration 5.	94
8.35 Computation times per simulation iteration for 8 experiment sets.	94
8.36 Computation times for a maximum setpoint boundary size of 1000 meters.	95
8.37 Computation times for a maximum setpoint boundary size of 2000 meters.	95
8.38 Forbidden area violation count with a setpoint boundary of 1000m.	96
8.39 Forbidden area violation count with a setpoint boundary of 2000m.	96
8.40 Computation time for three different control methods using 100 simulation iterations.	97
8.41 Area constraint violation count using 100 simulation iterations.	97
8.42 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 2, initial condition iteration 16.	98
8.43 Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 3, initial condition iteration 16.	98
8.44 Larger forbidden area, experiment set 1, initial condition iteration 3.	98
8.45 Larger forbidden area, experiment set 4, initial condition iteration 8.	98
8.46 Separation violation count for large forbidden area experiment.	99
8.47 Representation of comparison tests of three decentralized controller setpoint methods.	100
8.48 Constraint violations for the bi-level controller option 1, 2 and 3 and two versions of a single level controller.	100
B.1 Aircraft trajectory with static obstacle, with $H_p = 5$ and control sample time = 10 seconds.	107
B.2 Aircraft trajectory with static obstacle, with $H_p = 10$ and control sample time = 10 seconds.	107
B.3 Aircraft trajectory with static obstacle, with $H_p = 15$ and control sample time = 10 seconds.	107
B.4 Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 10 seconds.	107
B.5 Aircraft trajectory with static obstacle, with $H_p = 15$ and control sample time = 15 seconds.	108
B.6 Aircraft trajectory with static obstacle, with $H_p = 15$ and control sample time = 20 seconds.	108
B.7 Aircraft trajectory with static obstacle, with $H_p = 5$ and control sample time = 15 seconds.	108
B.8 Aircraft trajectory with static obstacle, with $H_p = 10$ and control sample time = 15 seconds.	108
B.9 Aircraft trajectory with static obstacle, with $H_p = 5$ and control sample time = 20 seconds.	108
B.10 Aircraft trajectory with static obstacle, with $H_p = 10$ and control sample time = 20 seconds.	108
B.11 Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 10 seconds.	109
B.12 Aircraft velocity with static obstacle, with $H_p = 20$ and control sample time = 10 seconds.	109

B.13 Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 10 seconds.	109
B.14 Aircraft velocity with static obstacle, with $H_p = 20$ and control sample time = 10 seconds.	109
B.15 Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 15 seconds.	109
B.16 Aircraft velocity with static obstacle, with $H_p = 20$ and control sample time = 15 seconds.	109
B.17 Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 15 seconds.	110
B.18 Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 20 seconds.	110
B.19 Aircraft velocity with static obstacle, with $H_p = 5$ and control sample time = 20 seconds.	110
B.20 Aircraft velocity with static obstacle, with $H_p = 10$ and control sample time = 20 seconds.	110
B.21 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 15$ and control sample time = 10 seconds.	110
B.22 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 20$ and control sample time = 10 seconds.	110
B.23 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 5$ and control sample time = 15 seconds.	111
B.24 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 10$ and control sample time = 15 seconds.	111
B.25 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 5$ and control sample time = 10 seconds.	111
B.26 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 10$ and control sample time = 10 seconds.	111
B.27 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 5$ and control sample time = 20 seconds.	111
B.28 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 10$ and control sample time = 20 seconds.	111
B.29 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 15$ and control sample time = 15 seconds.	112
B.30 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 15$ and control sample time = 20 seconds.	112
B.31 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 20$ and control sample time = 15 seconds.	112
B.32 Aircraft distance from nominal trajectory with static obstacle, with $H_p = 20$ and control sample time = 20 seconds.	112

List of Tables

8.1	Initial conditions for experiment 1, decentralized static obstacle avoidance	84
8.2	Results experiment 1, decentralized static obstacle avoidance.	84
8.3	Initial condition for Experiment 2, bi-level obstacle avoidance.	89
8.4	Experiment 2, prediction horizon and and control sample time settings.	89
8.5	Experiment sets for 100 iterations.	96
8.6	Experiment sets for testing the influence of setpoint method options 1, 2 and 3.	99
C.1	Decentralized and centralized Boeing 747 simulation parameters	113

List of Symbols - (Preliminary Report)

Δx_{ij}	Horizontal separation between aircraft i and j
Δx_{oj}	Horizontal distance between obstacle (o) and aircraft (j)
Δx_{sj}	Horizontal distance between setpoint (s) and aircraft (j)
Δy_{ij}	Vertical separation between aircraft i and j
Δy_{oj}	Vertical distance between obstacle (o) and aircraft (j)
Δy_{sj}	Vertical distance between setpoint (s) and aircraft (j)
δ_{MaxSep}	Maximum Euclidean distance from setpoint
δ_{MinSep}	Minimum required separation between obstacle and aircraft)
δ_{ij}	Separation distance between aircraft i and j
Δt_c	Control sample time centralized controller
Δt_d	Control sample time decentralized controller
ψ	Aircraft bank angle
ρ	Air density
φ	Aircraft heading angle
a	Aircraft acceleration
A_f	Fuel coefficient
B_f	Fuel coefficient
C_f	Fuel coefficient
C_L	Lift coefficient
$C_{D,0}$	Zero-lift drag coefficient
D	Drag force
D_{cpa}	Distance at closest point of approach
$D_{\text{rel},x}$	Relative horizontal distance
$D_{\text{rel},y}$	Relative vertical distance
D_{rel}	Relative Euclidean distance
F	Fuel flow
g_0	Gravitational acceleration
H_c	Control horizon
H_p	Prediction horizon
H_{p_c}	Centralized prediction horizon

H_{p_d}	Decentralized prediction horizon
J	Cost function value
k_0	Air density
m	Aircraft mass
n_{set}	Setpoint number
N_{eng}	Number of engines
S	Wing surface area
T	Thrust force
t_{cpa}	Time to closest point of approach
T_{ratio}	Thrust ratio
T_{static}	Static thrust force
v	Aircraft velocity
V_{avg}	Average velocity
V_i	Absolute velocity aircraft (i)
V_j	Absolute velocity aircraft (j)
V_0	Initial velocity
$V_{\text{rel},x}$	Relative horizontal velocity
$V_{\text{rel},y}$	Relative vertical velocity
V_{rel}	Relative velocity
W_x	Wind disturbance - horizontal direction
W_y	Wind disturbance - vertical direction
$W_{1,2,3}$	Cost function weight
x	Horizon aircraft position
x_{set}	Horizontal position of setpoint
x_{final}	Destination coordinate
y	Vertical aircraft position
y_{set}	Vertical position of setpoint
y_{final}	Destination coordinate
\mathbf{y}	Output vector
k	Discrete sample instance
t	time

Abbreviations

ASAS	Airborne Separation Assurance Systems
ABP	Achieve by Point
ABS-B	Automatic Dependent Surveillance Broadcast
ATC	Air Traffic Control
ATM	Air Traffic Management
BLC	Bi-level Controller
BL-RC	Bi-level Rough Controller
BL-DC	Bi-level Detailed Controller
CC	Centralized Controller
CD&A	Conflict Detection and Avoidance
CD&R	Conflict Detection and Resolution
CI	Cost-Index
CTAS	Center TRACON Automation System
DAG-TM	Distributed Air/Ground Traffic Management
DC	Decentralized Controller
ETA	Estimated Time of Arrival
ERASMUS	En-Route Air Traffic Soft Management Ultimate System
FF	Free-Flight
IFR	Instrument Flight Rules
IM	Interval Management
LQR	Linear Quadratic Regulator
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-Linear Programming
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
PHARE	Programme for Harmonized Air traffic Management in Europe
QDMC	Quadratic Dynamic Matrix Control
RHC	Receding Horizon Control
RNAV	Area Navigation
SL-AC	Single-level Accurate Controller
SL-DC	Single-level Detailed Controller
SL-RC	Single-level Rough Controller
SP	Setpoint

I

Scientific Paper

An Exploration of Integrated Bi-level Local and Global Model Predictive Control for Air Traffic Control Applications

D. Tol (MSc Student)

Supervisors: Dr. A. Jamshidnejad & Prof. dr. ir. J.M. Hoekstra

Section: Control & Simulation, Department: Control and Operations

Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands

abstract - A hierarchical bi-level model predictive controller is proposed in this thesis to reduce the computational complexity of controlling a large scale air traffic control problem, using a model predictive control approach. The bi-level controller developed and tested in this research project is a combination of a global, long-term, slow-rate, centralized model predictive controller and a local, short-term, fast-rate decentralized model predictive controller that aims to cooperatively guide aircraft towards their destination while avoiding forbidden areas. The bi-level controller is compared to both single-level controllers it is contrived to explore the benefits achieved by the cooperation of the individual controllers, in the context of an air traffic control application. An accurate baseline model predictive controller is used to compare the computational efficiency advantage gained when using the bi-level control structure. The bi-level controller proves to attain a superior control performance over both its contributing parts. The bi-level controller provides a more accurate control performance than the single global centralized controller and performs better in trajectory optimization than the local decentralized controller. Furthermore, the controller performance of the accurate baseline controller can be approached with the bi-level controller at a reduced computation time.

Keywords: model predictive control, MPC, air traffic control, ATC, hierarchical control, bi-level, centralized, decentralized, global, local

I. INTRODUCTION

A forecast by EUROCONTROL in 2018 predicts flight movements to grow by 53% in 2040 compared to 2017 [1]. An ever-increasing rise in the workload for air traffic controllers as well as further saturation of certain air spaces is imminent if innovations in air traffic control are not implemented. Reducing flight time, delays, and minimizing aircraft emissions are

TABLE I: Frequently used abbreviations

ATC	Air Traffic Control
BLC	Bi-level Controller
BL-DC	Bi-level Detailed Controller
BL-RC	Bi-level Rough Controller
LaT	Look-ahead Time
LoD	Level of Detail
RHC	Receding Horizon Control
SL-DC	Single-level Detailed Controller
SL-RC	Single-level Rough Controller
SL-AC	Single-level Accurate Controller
MPC	Model Predictive Control

other reasons to strive for more efficient Air Traffic Control (ATC).

In this thesis, a hierarchical bi-level controller using Model Predictive Control (MPC) is developed for air traffic trajectory optimization. The developed system is compared to multiple single-level MPC controller variants to investigate the benefits and drawbacks of the bi-level model predictive controller. An attempt will be made to uncover synergism between a global, slow-rate, centralized model predictive controller and a local, fast-rate, decentralized model predictive controller and the benefits of combining each controller in the overall performance of the control solution and the computational efficiency will be probed in the context of ATC. Table I provides an overview of the frequently used abbreviations and Table II shows the frequently used symbols in this thesis. When the subscript d is added to any of the symbols the detailed local controller is indicated and the subscript r is used to indicate the rough global controller.

Thesis Outline

First, background information on MPC, hierarchical structures and centralized vs. decentralized control is provided in Section II. The problem definition, as well as the main contributions of this thesis, are stated in Section III. The concept and optimization formulation of the bi-level model predictive controller is presented

TABLE II: Frequently used symbols

Symbol	Description	Unit
α	Acceleration	m/s ²
β	Heading change	deg/s
δ_{ac}	Aircraft to aircraft separation	km
δ_{obs}	Aircraft to obstacle separation	km
δ_s	Forbidden area radius	km
φ	Heading angle	deg
c	Control sampling time	s
c_d	Local control sampling time	s
c_r	Global control sampling time	s
c_s	Setpoint sampling time	s
k	Control step counter	-
k_d	Local control step counter	-
k_r	Global control step counter	-
k_s	Setpoint step counter	-
n_{set}	Setpoint index	-
r_s	Setpoint radius	km
\mathbf{u}	Controller input	-
v	Aircraft velocity	m/s
w	Wind velocity	m/s
\mathbf{x}	Aircraft position	km
\mathbf{x}_f	Destination position	km
\mathbf{x}_{set}	Setpoint position	km
J_r	Global cost function	-
J_d	Local cost function	-
N_{set}	Number of setpoints used	-
N_{ac}	Number of aircraft	-
N_p	Prediction horizon	-
W	Cost function weight	-

in Section IV, followed by an explanation of the case study setup in Section V. The simulation results are presented in Sections VI and VII. In these sections, eight different scenarios are described in detail to elaborate on multiple aspect of the controller. Following the simulation results, a discussion on the results and conceptual exploration is provided in Section VIII. Finally, the most important findings and prospects for future research are concluded in Section IX.

II. BACKGROUND

Several key concepts for this study are explained and supported by relevant literature in this section.

A. Model Predictive Control

In this research project, MPC is utilized for aircraft trajectory optimization. MPC is a control strategy capable of achieving an optimal sequence of control inputs that minimizes a cost function. MPC uses a model of the system dynamics to predict the future states of the system, given a series of control inputs. At the rate of the control sampling time, a sequence

of control inputs is determined, for a certain control horizon (N_c), within a given prediction horizon (N_p). The control horizon can be equal to, or smaller than the prediction horizon. When the optimization is completed, the first control input of the optimal input sequence is applied to the controlled system and the remainder of the input sequence is disregarded, the time window shifts one time step further and the optimization problem is repeated. A major advantage and key feature of MPC is the possibility to include constraints on the input, output, and states of the system that is being controlled.

Model predictive control, also referred to as Receding Horizon Control (RHC), has seen an increasing use in the past decades in many different fields of engineering, from automotive [2][3][4] to industrial processes [5][6], maritime [7] and aviation [8][9][10][11][12]. Research in next-generation air traffic control has also extended to drones and small unmanned vehicles and how these aircraft can safely operate without risking interference with air traffic [13][14][15]. In the survey paper by Garcia et al. from 1989 [16], an extensive history of early applications of MPC is described.

B. Centralized and Decentralized Control Structures

From a mathematical perspective, a centralized controller with global and perfect knowledge of the system could determine the global optimum. However, for large scale complex problems such as air traffic control, this can be computationally challenging.

Furthermore, the uncertain nature of air traffic prediction is a downside of global optimization, rendering the global optimization solution far from ideal. The low predictability of the problem demands near real-time updates of the optimization problem, which can be challenging due to the computational complexity of the optimization problem.

In order to reduce the computational complexity, a centralized control system can be divided into multiple smaller decentralized control systems. For air traffic control this concept is often referred to as 'free-flight', where each aircraft is free to determine its own optimal solution.

From the review by Al-Gherwi et al. [17], it is concluded that a centralized model predictive control unit with one overall cost function for the entire system can guarantee optimal performance. However, when the system is fully decentralized with independent local models and with independent local objectives functions for each subsystem, the performance could potentially become sub-optimal.

For a control system that manages air traffic separation, some level of cooperation by the different aircraft is desired. Within decentralized control systems, the knowledge that different aircraft have of each other is

not used to find the optimal solution cooperatively. When no centralized controller is present, an optimization sequence needs to be defined. A predefined sequence is proposed in [9], where the order of the optimization sequence is shown to have a substantial effect on the outcome. A distributed air traffic management controller is proposed in [18], however, this solution has a better result for low traffic densities.

In this thesis, centralized and decentralized control is combined to utilize the strengths of each method to reducing issues such as computation time of the centralized system and avoiding implementation of optimizations sequences on the decentralized control level.

C. Hierarchical Control Structures

In the review paper by Tatjewski [19], hierarchical systems and multilayered control problems using optimization on at least one level are discussed. Temporal, spatial, and functional decomposition of a system are mentioned to be the main methods for dividing control tasks into simpler sub-tasks in a hierarchical control system approach.

When a functional decomposition is applied, each level of the hierarchical control structure has different control objectives. The higher level can, for example, be a supervisory optimization-based controller and the lower level could be the direct control layer, which could be classical PID or a more advanced control algorithm such as MPC [19].

Temporal decomposition is described by Tatjeswki for a system where fast and slow state variables are present that change at a significant time difference. In a temporal decomposition, each sub-level can be functionally similar, yet the control sampling times and prediction horizons usually become longer and process models and disturbance dynamics tend to become slower when moving up in hierarchical levels [19].

In a hierarchical control system, setpoints can be used to dictate target states to lower level controllers. The higher level of the hierarchical control structure aims to find a global optimum, which is subsequently converted to setpoints, that act as references for the lower level controllers. Using setpoints in multi-leveled (MPC) structures is not new and is used and discussed extensively in [5][6][19][20], where a higher level MPC controller or different optimizer determines setpoints for a lower level tracking problem. The concept of setpoint optimization is used in this thesis, to integrate different control levels and introduce a level of cooperation between the controllers.

The novelty of this controller in this thesis is that the setpoints are also used as constraints for the lower level controllers, instead of the objectives and references.

In this thesis, all three methods for decomposition are explored in an air traffic control application. The main reason to apply a hierarchical control structure is to reduce the computational complexity of the controlled system. By utilizing different time frames for each controller, the system is separated into fast and slow dynamics, due to the introduction of different control sampling times. The spatial decomposition is used when controlling multiple aircraft on one level and controlling a single aircraft on the other level and manifests as centralized vs. decentralized control in this thesis. Finally, the functional decomposition is utilized by implementing different control objectives on each control level and by the division of a centralized controller and a decentralized controller. Realizing the control system into a hierarchical structure could potentially create a more efficient controller. Moreover, the introduction of two controllers can lead to conceptual advantages related to air traffic control which are discussed in Section VIII.

Another extensive overview of distributed and decentralized hierarchical architectures can be found in a review paper by R. Scattolini [21]. Scattolini explains that MPC can be used at any level of the controller to take full advantage of the input, state, and output constraints at each level. Despite this fact, not many applications or research have thus far been conducted where multiple hierarchical MPC controllers are combined [21].

One example that uses a bi-level optimization based controller is researched by D'Amato et al. [14]. In the context of drone formation flying and obstacle avoidance, a generic algorithm is combined with shortest path algorithms for UAV formation flight, collision avoidance, and forbidden area evasion in a bi-level control structure.

Another example of functional decomposition for air traffic control is researched by George J.Pappas [22]. Where a higher level controller gives commands to a lower level controller. The higher level controller uses a coarser model than the lower level controller. One of the main challenges is described to be, designing hierarchical system architecture is to achieve the comparability of the functionality and objectives of each control level [22]. Moreover, according to Scattolini, there is no systematic method to select the best (distributed) control strategy and these systems lack systematic design methods that guarantee well-assessed properties [21].

This thesis aims to contribute to the systematic design of a multi-level model predictive control by an exploration of several control strategies, and exploiting the potential of having multiple control levels in the context of air traffic control. Additionally, the concepts of centralized and decentralized control are combined in a hierarchical control structure in order to

guide multiple-aircraft towards their destination while maintaining adequate positional separation between individual aircraft.

III. PROBLEM DEFINITION

The popularity of MPC for optimal control applications can be ascribed to its ability to handle both physical and control constraints. Furthermore, MPC can be applied to control linear and non-linear systems with linear or non-linear models, constraints, or cost functions.

A centralized controller with global knowledge of the controlled system, using a large prediction horizon and a fast control input sampling time, can theoretically provide the best solution. The disadvantage of such a system is that the computation time can quickly become a limiting factor for fast real-time applications because the optimization algorithm is repeated at the control sampling time. The computational burden of MPC can increase drastically when the size of the problem grows. The number of agents, the controller sampling time, the prediction and control horizon, and the model complexity can all harm the real-time performance.

It is difficult to theoretically prove the stability and performance of non-linear MPC [17]. The optimal control problem requires a finite horizon to be able to be repeatedly solved online. However, to ensure stability, an infinite horizon is necessary [23]. The numerical burden of centralized MPC is often the main driving factor for the need for practical improvements. Decentralized or distributed MPC solutions have been proposed in order to reduce the computational burden of centralized MPC solutions in the context of UAVs by D'Amato et al. [13]. D'Amato et al. apply a prediction unit on the higher level to predict and resolve collisions based on the ICAO right of way rules and a decentralized lower level MPC is used for trajectory tracking of the controlled UAVs.

In [9] multiple different control levels are used for a decentralized air traffic approach, where a decentralized MPC is used at the higher controller level. To the best of the author's knowledge, a contribution can be made by exploring the possibilities of combining centralized and decentralized controller in a hierarchical control structure to improve the real-time performance.

Larger prediction horizons typically increase in the computation time required to optimize the controller input. One way to reduce the computational burden is, therefore, to reduce the prediction horizon while maintaining the controller sampling time, effectively reducing the look-ahead time. Consequently, a reduction in the quality of the control solution is expected, since only short-term knowledge of the system is used.

A different approach towards increasing computational efficiency is to increase the controller sampling time, naturally reducing the number of optimization variables within the look-ahead time. The accuracy of the solution is now compromised, due to the rough discretization and less frequent control sampling time. However, the controller does include future information that was lost in the previous method.

Since MPC uses a model of the system to predict future outputs, predicted states further into the future become increasingly unreliable due to the accumulation of errors, disturbances, and model uncertainties.

If MPC is applied to a system such as air traffic, the uncertain nature of global air traffic prediction can become a problem. The low predictability over the full length of a flight path is an issue rendering the global optimization solution far from ideal. The low predictability of the problem demands near real-time updates of the optimization problem, which is computationally practically impossible with the current technology. However, if the significance of future predictions is valued less and the lower far-time predictability is accepted and only used for general guidance, it can prove to be valuable information.

Main Contributions

The main contributions of this research are to investigate the integration of two model predictive controllers that cooperatively improve the collective results of the individual controllers for an air traffic control application. The novelty of the project lies in the combination of a global long term, slow-rate, model predictive controller with a local short term, fast rate, model predictive controller, to obtain a single coherent bi-level hierarchical controller. The short term fast rate MPC level only has local knowledge of the system, a smaller prediction horizon, with a fast control sampling time. The long term slow-rate MPC level operates with global knowledge of the system, a larger prediction horizon, and a slower control sampling time.

Among other goals, this research project aims to reduce the computational burden while minimizing the compromise to the control performance. Furthermore, the goal is to contribute to developing an understanding of how the potential of two control levels can be exploited, to reduce the computational complexity of an application such as air traffic. Finally, the shortcomings and important considerations that need to be made when designing a hierarchical bi-level model predictive controller will be explored and highlighted in the context of air traffic control based on several case studies and multiple scenarios.

IV. BI-LEVEL CONTROLLER SETUP

The Bi-level Controller (BLC) developed for this research project, is a combination of two model predictive controllers that are operating in a hierarchical structure. The leading controller is a global MPC with a large look-ahead time and a rough control input sampling time. This controller is called the 'Bi-level Rough Controller (BL-RC). The other MPC is a local short term controller, with a shorter look-ahead time operating with a shorter control period. This controller is hereafter called the 'Bi-level Detailed Controller' (BL-DC) and serves as the direct control layer. The local BL-DC and global BL-RC combined form the BLC.

Since each controller operates at a different pace and at different time-frames, they each have their strengths and weaknesses, see Section III. When the two computation time reducing methods are applied in a bi-level controller structure, the compromise between accuracy and computation time can be avoided. In Table III the benefits and drawbacks of each method are summarized.

TABLE III: Benefits (+) and drawbacks (-) of the different controller variations in terms of Look-ahead Time (LaT) and Level of Detail (LoD).

	LaT	LoD
Global rough controller	+	-
Local detailed controller	-	+
Bi-level controller	+	+

The BL-RC reduces the computation time by applying a larger control sampling time (c_r), while keeping the look-ahead time constant, sacrificing accuracy as a result of a coarse discretization. The loss in detail is regained by the introduction of the local model predictive controller, with a smaller control sampling time (c_d) and thus a finer discretization. The BL-DC gains its computational efficiency by reducing the look-ahead time, for which the control input is optimized. The long-term information that is lost in this process, is in turn provided by the BL-RC. For short term predictions of the solution, a certain level of detail needs to be achieved which is considered a design requirement. The level of detail will be rougher for future predictions outside the look-ahead time of the detail controller.

In Figure 1, the level of detail of both controllers is illustrated over a time horizon. It can be seen that the local detailed controller updates its control input more frequently than the global controller. However, the local controller determines a sequence of control inputs for a shorter look-ahead time with a prediction horizon of Np_d . The global rough controller has a

larger look-ahead time with a prediction horizon of Np_r . These two controllers together form the BLC and provide the required level of detail until Np_d . After the local detailed controller's look-ahead time the level of detail of the control solution is reduced. Nevertheless, a reduction of detail for far ahead predictions is acceptable if the additional information does not contribute to a significant improvement of the solution. Due to the low predictability of global air traffic for far ahead predictions, a highly detailed control solution might be unreliable. The BLC, therefore, places more emphasis on optimizing the short term control solution. The receding horizon properties of MPC makes sure that the time frame that contains adequate detail shifts forward with every new iteration. The detailed controller inputs with a longer look-ahead time can also be seen in Figure 1, indicated by the yellow dashed line. This part of the control solution is lost when using the BLC.

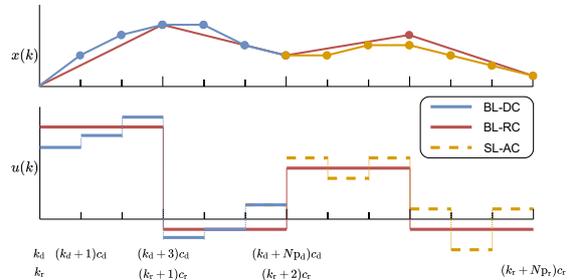


Fig. 1: Different controller sampling periods and prediction horizons resulting in different levels of detail for different look-ahead times.

The number of aircraft that is controlled by each control level, is another important distinction creating an additional categorical division of the controller levels based on centralized and decentralized control. The global BL-RC gains a supplementary task of aircraft separation and pursuing a global optimum. The local BL-DC controllers are placed on the direct control level, which is only tasked by creating an optimal path for one single aircraft, without any knowledge of the global situation and neighboring aircraft, by using instructions from the BL-RC.

A. BLC Concept

The Bi-level Rough Controller (BL-RC) finds a long term conflict-free trajectory for each aircraft in the system, based on an optimal sequence of control inputs. The predicted states of the BL-RC are converted to setpoints. The coordinates of the setpoints are determined using the aircraft model and the optimized control inputs sequence, provided by the BL-RC. The predicted setpoint coordinates are thereafter converted to circular area constraint and subsequently applied as dynamic constraints to the local detailed controllers.

The setpoints are not specifically used as references for a tracking task since the setpoints are determined using a limited amount of detail. Therefore, the local detailed controller (BL-DC) is granted the authority, to improve the solution in the short term and deviate from the setpoint within a safe margin. The setpoints are updated at the rate of the BL-RC control interval.

Figure 2, shows a schematic representation of the BLC architecture. The addition of multiple aircraft

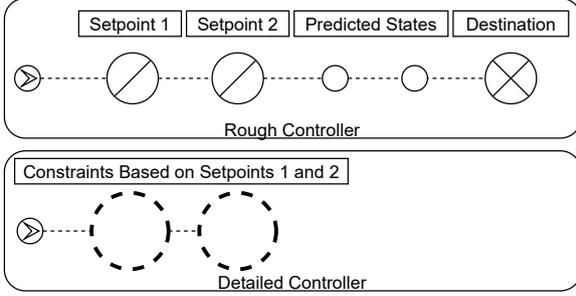


Fig. 2: Schematic representation of the BLC. The setpoints determined by the global BL-RC are converted to circular area constraints for the local BL-DC.

increases the number of optimization variables for the centralized controller level. However, this can remain computationally feasible due to the rough sampling time. By not including a link between aircraft on the local decentralized controller level there is no increase in optimization complexity of this controller level, regardless of the number of aircraft in the system if parallel computing is utilized.

B. Optimization Formulation

In this section, the method of combining the rough and detailed controllers is explained by defining and formulating the optimization problem for both controllers and the link to integrate the controllers. Furthermore, the underlying model and the BLC algorithm are presented.

The aircraft model is a set of kinematic equations used to compute the position (x, y) , velocity (v) and heading (φ) under a certain wind velocity (w) given by,

$$\dot{x} = v \cdot \sin(\varphi) + w_x \quad (1)$$

$$\dot{y} = v \cdot \cos(\varphi) + w_y \quad (2)$$

$$\dot{v} = \alpha \quad (3)$$

$$\dot{\varphi} = \beta \quad (4)$$

where the acceleration or deceleration input (α) and the heading change (β) form the manipulated variables.

Global Bi-Level Rough Controller

The bi-level controller operates at two different time

frames with different discretized sampling periods. The global BL-RC advances with discrete time steps, where two consecutive time steps are a sampling time period of c_r seconds apart and k_r gives the current time step of the rough controller. At each control time step $k \in \{k_r, \dots, k_r + N_{p_r} - 1\}$, a control input is determined resulting in the following control input sequence for the global BL-RC at k_r ,

$$[\mathbf{u}_r(k_r), \mathbf{u}_r(k_r + 1), \dots, \mathbf{u}_r(k_r + N_{p_r} - 1)]^T \quad (5)$$

where the discrete controller inputs, $\mathbf{u}_r = [\alpha, \beta]$, are optimized for each aircraft such that the control sequence (Eq. 5) minimizes the global cost function (J_r), which is given by the expression below,

$$\begin{aligned} J_r(k_r) = & W_{r_f} \cdot \sum_{j=1}^{N_{ac}} \|\mathbf{x}_{r,j}(k_r + N_{p_r}) - \mathbf{x}_{f,j}\|_2 \\ & + W_{r_{\alpha 1}} \cdot \sum_{j=1}^{N_{ac}} \sum_{k=k_r}^{k_r + N_{p_r} - 1} |\alpha_{r,j}(k) - \alpha_{r,j}(k-1)| \\ & + W_{r_{\beta 1}} \cdot \sum_{j=1}^{N_{ac}} \sum_{k=k_r}^{k_r + N_{p_r} - 1} |\beta_{r,j}(k) - \beta_{r,j}(k-1)| \\ & + W_{r_{\alpha 2}} \cdot \sum_{j=1}^{N_{ac}} \sum_{k=k_r}^{k_r + N_{p_r} - 1} |\alpha_{r,j}(k)| \\ & + W_{r_{\beta 2}} \cdot \sum_{j=1}^{N_{ac}} \sum_{k=k_r}^{k_r + N_{p_r} - 1} |\beta_{r,j}(k)| \end{aligned} \quad (6)$$

where $\alpha_{r,j}$ and $\beta_{r,j}$ at $k-1 = 0$ are assumed to be 0. The optimization is subjected to the following constraints, at each control time step $k \in \{k_r, \dots, k_r + N_{p_r} - 1\}$:

$$\mathbf{x}_r(k+1) = f_r(\mathbf{x}_r(k), \mathbf{u}_r(k)) \quad (7)$$

$$\mathbf{u}_{r,j}^{\min} \leq \mathbf{u}_{r,j}(k) \leq \mathbf{u}_{r,j}^{\max} \quad (8)$$

$$v_{r,j}^{\min} \leq v_{r,j}(k+1) \leq v_{r,j}^{\max} \quad (9)$$

$$\|\mathbf{x}_{r,j}(k+1) - \mathbf{x}_{obs}\|_2 > \delta_{obs} \quad (10)$$

The terminal part of the cost function associated with the weight W_{r_f} is used to minimize the relative distance of the aircraft position (\mathbf{x}_j) to the location of the destination $(\mathbf{x}_{f,j})$ at the time instance $(k_r + N_{p_r})c_r$. While the stage part of the cost function associated with the weights, $W_{r_{\alpha}}$ and $W_{r_{\beta}}$, penalizes the maximum absolute accelerations and heading changes in order to have a smooth trajectory and improve passenger comfort, within the prediction time window.

In case more than one aircraft is included in the simulation, the rough optimization is subjected to the following aircraft separation (δ_{ac}) constraint.

$$\|\mathbf{x}_{r,i}(k+1) - \mathbf{x}_{r,j}(k+1)\|_2 > \delta_{ac} \quad (11)$$

for $k \in \{k_r, \dots, k_r + N_{p_r} - 1\}$, where the subscripts i and j are aircraft indices and $i \neq j$.

Setpoint Determination

The optimized control input sequence determined at k_r , (Eq. 5) can be used to compute a series of predicted coordinates placed at the BL-RC controller sampling rate (c_r) within the time interval, $[(k_r + 1)c_r, (k_r + N_{p_r})c_r]$. These coordinates can serve as setpoints for the local BL-DC controllers.

Since the two controller levels are interconnected through the setpoints and the setpoint constraints, the setpoints must fall within the look-ahead time of the local detailed controller. The prediction horizon of the local BL-DC is dependent on the ratio of controller sampling times and number of setpoints used by,

$$N_{p_d} \geq N_{set} \cdot \frac{c_r}{c_d} \quad (12)$$

where N_{set} is given by the number of setpoint constraints used by the BLC. If the controller sampling ratio (c_r/c_d) is relatively large, the prediction horizon (N_{p_d}) will consequently increase. This effectively increases the computation time for the optimization algorithm on the detailed controller level, increasing the total computation time of the BLC.

In order to decrease the computation time on the local controller level the setpoints can be placed at a faster rate, by reducing the setpoint time interval. The intermediate discrete coordinates resulting from the model and the optimized controller inputs of BL-RC can be subdivided further into a setpoint time sampling time (c_s), where c_s is an integer factor of the rough control sampling time (c_r). The setpoint sampling time must be equal or smaller than the rough control sampling time. The setpoints coordinates (\mathbf{x}_{set}) are determined using the aircraft model and the optimized control input sequence (Eq. 5), implemented as a zero order hold piece wise continuous control input. The prediction horizon N_{p_d} can then be found in accordance to the following relation,

$$N_{p_d} \geq N_{set} \cdot \frac{c_r}{c_d} \cdot \frac{c_s}{c_r} \quad (13)$$

It should be noted that when intermediate setpoints are utilized ($c_s < c_r$), the setpoints coordinates are computed using the same control input until the next discrete moment in time at $(k_r + 1)c_r$, when the next control input becomes available, every time when,

$$\text{mod} \left(\frac{c_s}{c_r} k_s, 1 \right) = 0 \quad (14)$$

where k_s indicates the setpoint counter. The position of the center of the dynamic circular setpoints are determined by the BL-RC as follows,

$$\mathbf{x}_{set}(n_{set}|k_r) = f_r \left(\mathbf{x}_r(n_{set} - 1|k_r), \mathbf{u}_r(n_s|k_r) \right) \quad (15)$$

where $n_{set} \in \{1, 2, \dots, N_{set}\}$, indicates the setpoint number and N_{set} gives the number of setpoints used in total. Throughout this thesis $N_{set} = 2$. Furthermore, the setpoints are computed with respect to the already realized state at $|k_r$. The term, n_s , in Equation 15, is given by the following equation and rounded down to the nearest integer,

$$n_s = n_{set} \frac{c_s}{c_r} - 1 \quad (16)$$

which makes sure the same control input (from Eq. 5) is used until the moment in time when the new control input becomes available.

It must be noted that, the computation of intermediate setpoints adds to the number of operations by the BLC controller, increasing the computational effort. Nevertheless, the added computations do not increase the computational burden of the optimization algorithms and are likely to yield a net reduction in computational effort because the intermediate setpoints are not part of the global optimization. The fact that the intermediate setpoints are a result of interpolation using the model and the previously determined optimal rough control inputs, they might be sub-optimal. Additionally, due to non-linearly of the system the intermediate setpoint might not be valid and result in large deviations and unwanted effects. Furthermore, because the intermediate setpoint have not been computed by means of an optimization algorithm they are not explicitly subjected to the optimization constraints. Therefore, utilizing intermediate setpoints like described in this section must be handled with care.

In this thesis intermediate setpoints are utilized only to reduce the prediction horizon of the local BL-DC and subsequently decrease the computation time of the BLC. When using a non-linear model. Future research could focus on a systematic approach to determine the ratio and values for c_r , c_d and c_s , which could be updated dynamically throughout the simulation. Determining the optimal values for the different controller sampling times is out of the scope of this thesis. In this project the values for c_r , c_d and c_s have been iteratively chosen for each case-study.

Local Bi-Level Detailed Controller

The local BL-DC advances with discrete time steps (k_d), with a sampling period of c_d seconds. At each control time step (k), a control input is computed to achieve a control input sequence for the BL-DC at k_d for $k \in \{k_d, \dots, k_d + N_{p_d} - 1\}$,

$$[\mathbf{u}_d(k_d), \mathbf{u}_d(k_d + 1), \dots, \mathbf{u}_d(k_d + N_{p_d} - 1)]^T \quad (17)$$

where the discrete controller input, $\mathbf{u}_d = [\alpha, \beta]$, is optimized such that the control input sequence at

discrete time steps minimizes the rough cost function (J_d) with cost function weights (W_d),

$$\begin{aligned}
J_d(k_d) = & W_{d_f} \cdot \|\mathbf{x}_d(k_d + N_{p_d}) - \mathbf{x}_f\|_2 \\
& + W_{d_{\alpha 1}} \cdot \sum_{k=k_d}^{k_d+N_{p_d}-1} |\alpha_d(k) - \alpha_d(k-1)| \\
& + W_{d_{\beta 1}} \cdot \sum_{k=k_d}^{k_d+N_{p_d}-1} |\beta_d(k) - \beta_d(k-1)| \\
& + W_{d_{\alpha 2}} \cdot \sum_{k=k_d}^{k_d+N_{p_d}-1} |\alpha_d(k)| \\
& + W_{d_{\beta 2}} \cdot \sum_{k=k_d}^{k_d+N_{p_d}-1} |\beta_d(k)|
\end{aligned} \tag{18}$$

where $\alpha_{d,j}$ and $\beta_{d,j}$ at $k-1=0$ are assumed to be 0. The optimization is subject to the following constraints at each control step $k \in \{k_d, \dots, k_d + N_{p_d} - 1\}$:

$$\mathbf{x}_d(k+1) = f_d(\mathbf{x}_d(k), \mathbf{u}_d(k)) \tag{19}$$

$$\mathbf{u}_{d,j}^{\min} \leq \mathbf{u}_{d,j}(k) \leq \mathbf{u}_{d,j}^{\max} \tag{20}$$

$$v_{d,j}^{\min} \leq v_{d,j}(k+1) \leq v_{d,j}^{\max} \tag{21}$$

$$\|\mathbf{x}_{d,j}(k+1) - \mathbf{x}_{\text{obs}}\|_2 > \delta_{\text{obs}} \tag{22}$$

The local BL-DC is also subjected to one or multiple setpoint constraints,

$$\|\mathbf{x}_d(k) - \mathbf{x}_{\text{set}}(k_s + n_{\text{set}})\|_2 \leq \delta_s \tag{23}$$

for $n_{\text{set}} \in \{1, 2, \dots, N_{\text{set}}\}$ and the setpoint number is counted with respect to the setpoint counter k_s . Equation 23 is not valid for each $k \in \{k_d, \dots, k_d + N_{p_d} - 1\}$ like the other constraints. The setpoints are computed by the global BL-RC and placed at larger discrete time steps than the time steps of the BL-DC. Therefore, the constraint can only be implemented at the RC-DC iteration when $k \cdot c_d$ for $k \in \{k_d + 1, \dots, k_d + N_{p_d}\}$ is coincident in time with the corresponding setpoint, which happens only when,

$$k \cdot c_d = (k_s + n_{\text{set}}) \cdot c_s \tag{24}$$

for $n_{\text{set}} \in \{1, 2, \dots, N_{\text{set}}\}$. To determine when the counter (k) on the range $k \in \{k_d + 1, \dots, k_d + N_{p_d}\}$ satisfies Equation 24, the following expression is used,

$$k_d + h = (k_s + n_{\text{set}}) \cdot \frac{c_s}{c_d} \tag{25}$$

Thus the setpoint constraint (Eq. 23) is applied for $k \in \{(k_s + n_{\text{set}}) \cdot \frac{c_s}{c_d} - h, \dots, (k_s + N_{\text{set}}) \cdot \frac{c_s}{c_d} - h\}$. Since $k_s c_s = k_d c_d$, the previous range for which Equation 25 is valid can also be written as $k \in \{k_d + n_{\text{set}} \cdot \frac{c_s}{c_d} - h, \dots, k_d + N_{\text{set}} \cdot \frac{c_s}{c_d} - h\}$. For this approach it is important that the ratio (c_s/c_d) is an integer value.

As a result of the receding horizon properties of the BL-DC and the different sampling times for both controllers, the setpoint constraint has to be applied at a different discrete instance of the prediction horizon. Therefore the counter (h) is increased by 1 every time k_d is increased by 1. Since h is a counter linking the controllers by applying the setpoint constraint to the appropriate instance in time, the iterator h is reset to 0 once the first setpoint is reached every time when,

$$\text{mod} \left(h, \frac{c_s}{c_d} \right) = 0 \tag{26}$$

Once the first setpoint has been reached a new sequence of setpoints is determined by a reiteration of the global controller or by selecting the next setpoints from the already determined sequence.

Finally, the first controller input from the optimal sequence (Eq. 17) is applied to each respective aircraft at k_d , on the time frame $[k_d c_d, (k_d + 1) c_d]$.

Algorithm 1 provides a high level overview of the working principle of the controller.

Algorithm 1 Bi-level Controller Algorithm.

```

1: for  $k_r$  is 1 to simulation time do
2:    $z = 0$ 
3:    $\mathbf{x}_s(z) = \mathbf{x}_r(k_r)$ 
4:   BL-RC minimize  $J_r$ 
5:   To find  $[\mathbf{u}_r(k_r), \mathbf{u}_r(k_r + 1), \dots, \mathbf{u}_r(k_r + N_{p_r} - 1)]^T$ 
6:   for  $n$  is 1 to  $N_{p_r}$  do
7:     for  $s$  is 1 to  $\text{int}(c_r/c_s)$  do
8:        $\mathbf{x}_s(z+1) =$ 
9:          $f_r(\mathbf{x}_s(z), \mathbf{u}_r(k_r + n - 1))$ 
10:       $z = z + 1$ 
11:     end for
12:   end for
13:    $\mathbf{x}_{\text{set}}[1 : z] = \mathbf{x}_s[1 : z]$ 
14:   for  $i$  is 1 to  $\text{int}(c_r/c_s)$  do
15:      $\mathbf{x}_{\text{set}}[1, z] = \mathbf{x}_{\text{set}}[i, z]$ 
16:      $h = 0$ 
17:     for  $k$  is 1 to  $\text{int}(c_s/c_d)$  do
18:        $h = k - 1$ 
19:       for  $j$  is 1 to  $N_{ac}$  do
20:         BL-DC minimize  $J_d$ 
21:         To find  $[\mathbf{u}_{d,j}(k_d), \mathbf{u}_{d,j}(k_d + 1), \dots, \mathbf{u}_{d,j}(k_d + N_{p_d} - 1)]^T$ 
22:          $\mathbf{x}_{d,j}(k_d + 1) =$ 
23:            $f_d(\mathbf{x}_{d,j}(k_d), \mathbf{u}_{d,j}(k_d))$ 
24:       end for
25:        $k_d = k_d + 1$ 
26:     end for
27:      $k_s = k_s + 1$ 
28:   end for

```

In Algorithm 1, z is an auxiliary iterator used when intermediate setpoints are applied (if $c_r/c_s \neq 1$). Line 17 of Algorithm 1 is explained by Equations (24-26). Notice on line 18 of Algorithm 1, how the detailed optimization is performed for each aircraft sequentially. This creates an analogy with decentralized control, where the detailed control loop could be executed on different computers, or separately by every aircraft in the system. Once a control input is determined for every aircraft, the simulation advances such that all aircraft receive their control inputs before the simulation continues. Since the aircraft on the decentralized control level are not coupled, the sequential optimization yields the same result as a parallel optimizing technique. Therefore, the sequential optimization only affects multi aircraft systems, which are presented in Section VII. Line 27 is included when feedback is used to determine the starting states for the new optimization instance of the BL-RC. If no measurement update is implemented the first setpoint at ($z = 1$) is used as the next state for the global BL-RC. This is only valid when the prediction model is equal to the system model and there are no disturbances. The optimization is performed using the MATLAB *fmincon* function with the *sqp* algorithm.

V. CASE STUDY SETUP

In this section, the case studies used to test the Bi-level Controller (BLC) performance are introduced. The case studies are divided into two main parts. In the first part, a series of static forbidden area avoidance tests, using one aircraft are presented. The forbidden areas are a combination of circles with equal radii. This first part is used to explore the working principles of the controller, provide an insight into the gain in computational efficiency, and to present the intricacies that arise from different approaches to the BLC structure.

In the second part, multi-aircraft simulations are performed and discussed. These scenarios serve to explore the possibilities for an extension into the multi-agent domain using the BLC and provide an introduction for future research.

A. Control Methods

Four controller versions are used during the case-studies for comparison purposes. Both of the control levels, that create the bi-level controller are individually capable of controlling the aircraft towards its destination. When either of the isolated controllers is used to control the aircraft they are referred to as 'Single-level Rough Controller' (SL-RC) or 'Single-level Detailed Controller' (SL-DC). The only difference between the single and bi-level detailed controllers is that the single-level detailed controller does not have the setpoints constraint implemented.

One other controller setup is used during the following case studies, which is the 'Single-level Accurate Controller' (SL-AC). The SL-AC operates at the look-ahead time of the global rough controllers (BL-RC and SL-RC) using the shorter sampling time of the detailed controllers (BL-DC and SL-DC). Since the look-ahead time of the SL-AC is equal to that of the BL-RC, the prediction horizon of the SL-AC can be computed using,

$$Np_a = \frac{c_r}{c_d} \cdot Np_r \quad (27)$$

The SL-AC is the controller of choice if real-time performance and computation time are of no concern. The solution found by the SL-AC serves as a baseline and is considered to be the desired end result that should be approached by the BLC. The computation time of the BLC is compared to the computation time required by the SL-AC.

B. Assumptions

- 1) All initial aircraft conditions are known.
- 2) The states of each aircraft are accurately measured, without noise or delay.
- 3) Disturbances are neglected.
- 4) The BL-DC only knows the initial conditions of the aircraft it controls.
- 5) The local detailed controllers do not know the states of other aircraft.
- 6) There is no communication delay between the BL-RC and BL-DC.
- 7) Computation delay is neglected during the simulations.
- 8) The BL-RC and BL-DC have accurate knowledge of the destinations and all forbidden areas.
- 9) All aircraft are limited to flight movements in the horizontal two-dimensional plane.
- 10) Aircraft mass is constant during the simulations.

C. Performance Metrics

Three performance metrics are used to assess the BLC's performance. The first metric is a high level assessment of the general trajectory that is determined by the controller. For this assessment, it is evaluated if the controller was capable of finding a solution in the right direction. The second metric is the level of detail that is achieved by the controllers, this metric is determined based on the control sampling time and the flight speed of the aircraft. The final performance metric is computational efficiency. The simulations are repeated four times, each time using one of the four control methods.

Level of Detail

To assess the level of detail, the main focus is on the BLC and both single-level controllers that it is contrived of. As a result of the discretization and linear

piece-wise trajectory approximation, the aircraft can trespass the forbidden area in between two evaluation samples. The level of detail or accuracy of the controller is based on the maximum theoretical forbidden area constraint violation. Since the forbidden areas are circles, the maximum infraction is directly related to the sampling time of the respective control method used (c), forbidden area radius (δ_s), and flight velocity (v). The analytic solution to the maximum violation (x) can be computed using the following relation,

$$x = \delta_s - \sqrt{\delta_s^2 - \left(\frac{1}{2}v \cdot c\right)^2} \quad (28)$$

This equation can be derived using the geometrical relations presented in Figure 3.

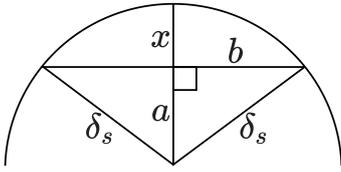


Fig. 3: Trigonometric relations to determine maximum analytic violation of forbidden area constraint (x), where $\delta_s = a + x$, $b = \frac{1}{2}v \cdot c$.

When flying at a constant velocity the maximum theoretical infringement x is presented as a function of the sampling time in Figure 4. It can be seen that the maximum forbidden area violation rises for larger controller sampling times until a maximum violation is reached. This happens when x is equal to the forbidden area radius (δ_s). At which point the circular area can be placed in between two coordinate points completely, allowing the aircraft to fly straight across a forbidden area.

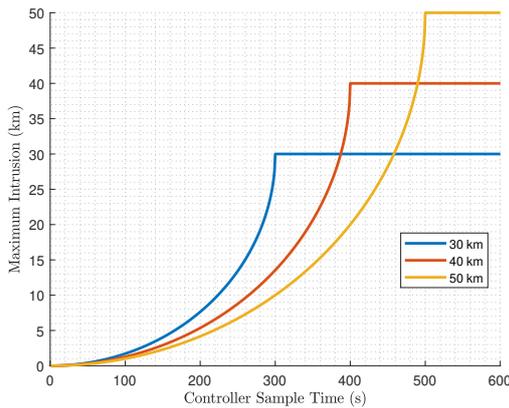


Fig. 4: Maximum analytic forbidden area violation of forbidden areas with radii of 30, 40 and 50 km for varying controller sampling times.

The relationship between velocity and computation time for a given forbidden area with a radius of 50

km can be seen in Figure 5. The controller sampling time has a large effect on the maximum area violation. This concept is what defines the level of detail of the controllers.

The Single-level Accurate Controller (SL-AC) operates at the same level of detail as the Bi-level and Single-level Detailed Controllers (BL-DC and SL-DC), because it uses the same controller sampling time.

When tuning the control sampling time of the Bi-level Rough Controller (BL-RC), the sampling time must be small enough such that the potential infeasible regions, such as the forbidden areas should be captured by the controller dynamics.

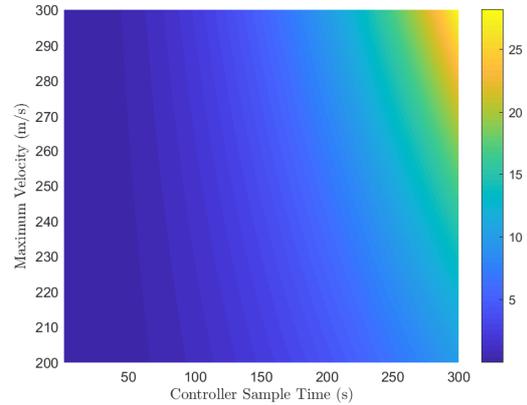


Fig. 5: Maximum analytic forbidden area violation (in km) of a forbidden area with a 50 km radius for various velocities and controller sampling times.

Considering that the setpoints for the BL-DC are determined by the BL-RC, the largest possible constraint infringement by the leading controller must be accounted for. When $c_s < c_r$, the intermediate setpoints can fall entirely within the forbidden area if the setpoint radius (r_s) is too small. Figure 6 shows one setpoint where r_s is large enough to provide a feasible region for the BL-DC (green area). Consequently, Equation 28 can also directly be used to determine the minimum setpoint radius (r_s). To determine the setpoint radius the maximum possible flight velocity should be used since it can be concluded from Figure 5 that a higher velocity results in a larger forbidden area violation. When the setpoint falls exactly on the midpoint of the line segment through a forbidden area the maximum violation must be less than the setpoint radius to allow for a feasible region within the circular setpoint constraint.

Computational Performance

The final performance metric is the relative computational efficiency. To determine the computational efficiency, the BLC is compared to the SL-AC baseline. The computational burden of the SL-RC and SL-

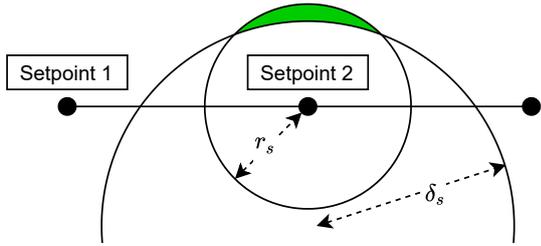


Fig. 6: Required setpoint radius to provide an feasible region for the BL-DC, when intermediate setpoints are used.

DC are not considered explicitly since they together contribute to the total computation time of the BLC.

For the first three scenarios and the multi aircraft simulation presented in part 2, the relative total computation time reduction that is reported is defined as,

$$\text{Reduction} = \frac{T_{\text{SL-AC}} - T_{\text{BLC}}}{T_{\text{SL-AC}}} \cdot 100\% \quad (29)$$

Scenario 4 is used to quantify the computational efficiency. Here the mean computation time for each iteration and the maximum computation times during the simulation are assessed in detail, for the BLC and the SL-AC. The influence of tuning the BLC, compared to the desired baseline is presented. A preliminary conclusion will be given to provide insight into the real-time performance.

VI. SINGLE AIRCRAFT SIMULATION RESULTS

For the first series of case studies, four scenarios with a single aircraft are presented in order to isolate the effect of the cooperation of different time-frame controllers. One aircraft starts at the position (0,0) km and flies towards its destination at (2500,0) km around or through a series of circular forbidden areas. Since the destination is far out of reach during the simulation time it is therefore not included in the trajectory figures, except for scenario 1 for illustration purposes. Furthermore, the simulation time steps are the same for all controller methods for each simulation in the same scenario. The simulation time steps are either 1 second or 5 seconds and specified in Appendix A.

A. Scenario 1: Static Obstacle Avoidance

The first scenario contains one aircraft that must fly around two sets of forbidden areas, to reach its destination. In the first scenario, the cost function weights are equal for all controllers. Scenario 1 is used to evaluate how the basic version of the Bi-level Controller (BLC) compares to each of the single-level controllers in terms of the achieved level of detail regarding the solution accuracy and the computational efficiency.

The constraint boundaries are similar for all the controller variations for the maximum allowed heading change. Furthermore, the controllers are not allowed to change the aircraft velocity, only the heading angle is controlled in order to change the position. The controller parameters and cost function weights, as well as the scenario setup, can be found in Appendix A. Furthermore, it should be noted that the setpoints for the Bi-level Detailed Controller (BL-DC) are determined at a faster rate of 120 seconds than the 240 seconds of the Bi-level Rough Controller (BL-RC) sampling time. Reducing the required N_{p_d} by a factor 2.

BLC, SL-RC, and SL-DC Trajectory Comparison

In Figure 7, the trajectories of the Single-level Rough Controller (SL-RC), Single-level Detailed Controller (SL-DC), and the BLC are portrayed. When observing the full zoomed-out overview of the general trajectories, the advantage of the SL-RC over the SL-DC becomes apparent. It can be seen that the SL-DC initiates the required heading change at a later stage compared to the SL-RC. This can be expected from the larger look-ahead time of the SL-RC of 16 minutes, compared to the 4 minute look-ahead time of the SL-DC. The forbidden areas are consequently captured within the look-ahead time earlier. The SL-RC is, therefore, able to change the control input sooner, reducing the required control effort.

This effect is also seen when evaluating the BLC, which is a combination of the SL-DC and SL-RC. Even though the direct control level of the BLC is similar to the SL-DC, the required heading change is initiated earlier. The aircraft is 'forced' into the right direction by the setpoint constraints. The combined system finds a more efficient trajectory than the SL-DC, in terms of path length. Therefore, it can already be concluded that the information provided by the global BL-RC to the local BL-DC yields a positive effect in terms of trajectory optimization.

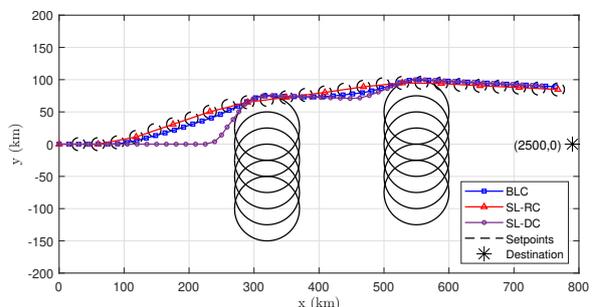


Fig. 7: Trajectory comparison of the BLC and the contributing SL-RC and SL-DC for scenario 1.

The strength of the SL-DC and how it in turn can also complement the SL-RC becomes visible when

taking a closer look at the trajectories, in Figure 8. Here the loss of detail as a result of the rough discretization is visible. Although the SL-RC did not violate any constraints during the optimization process, a relatively large 'shortcut' is taken through the forbidden areas. The SL-RC and BL-RC can violate the constraint with an analytical maximum of 10 km while the SL-DC, BL-DC, and SL-AC have a maximum theoretical violation of only 0.56 km, according to Equation 28. The level of detail of the solution is therefore increased by almost 95% when using the BLC, due to its shorter control sampling time.

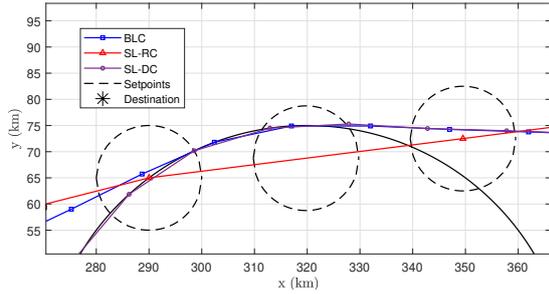


Fig. 8: Close up of Figure 7, showing the trajectory around a forbidden area.

The detailed part of the BLC, as its name implies, is better capable of accurately flying around forbidden areas. This part of the controller makes sure the required level of detail is maintained, in the short-term, which is lost when using the SL-RC.

Forbidden Area Violation Assessment

The theoretical maximum violation of the forbidden area has already been discussed in the previous section, the actual simulated violation is quantified in Figure 9. From this simulation, it is apparent that for the BLC, SL-AC, and the SL-RC, all the coordinates at which the constraints are evaluated fall outside the forbidden area.

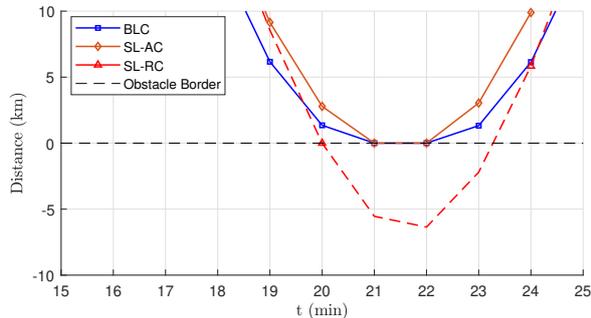


Fig. 9: Relative positional distance to the forbidden area boundary for scenario 1.

Nevertheless, it can be verified that the aircraft trespasses the forbidden area by more than 5 km

when using the SL-RC. Since the evaluation of the constraint is done at the sampling time of the BL-DC, the simulated violation by the BL-DC is nearly zero.

BLC and SL-AC Trajectory Comparison

A comparison of the Bi-level Controller (BLC) with the Single-level Accurate Controller (SL-AC) trajectory is presented in Figure 10. This comparison is made to assess the computational efficiency as well as the two different trajectories.

As is explained in Section V, the SL-AC is a controller version where no compromise is made between the prediction horizon and controller sampling time. From this comparison, it is found that the computation time for the whole simulation was reduced by 85% when using the BLC instead of the SL-AC.

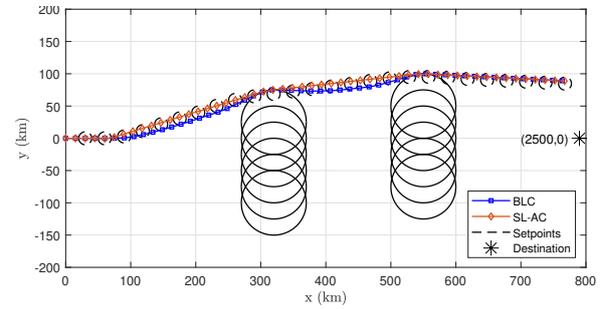


Fig. 10: Trajectory comparison of the BLC and the SL-AC for scenario 1. The setpoints are determined by the BL-RC and do not correspond to the SL-AC.

Even though the trajectories of the BLC and the SL-AC are relatively similar, a curve is visible in the BLC trajectory before the first row and second row of forbidden areas. This effect can be seen in the close-up in Figure 11.

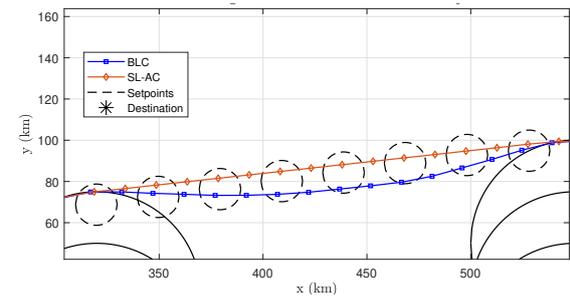


Fig. 11: Close up of Figure 10 between the two forbidden area, showing a curvature in the BLC path. The setpoints are determined by the BL-RC and do not correspond to the SL-AC.

The cost function of the BL-DC is tuned to reduce the distance between the aircraft and its destination. Therefore, minimizing the local cost function results in a heading change initiated by the BL-DC, for

as far as the setpoint constraints allow. The optimal solution found from a local perspective is to increase the heading angle in between the two forbidden area sets, which is not the optimal solution from a global perspective. This is caused by the fact that the final destination is at $y = 0$ km and the BL-DC is still unaware of the second set of forbidden areas due to its shorter look-ahead time. A larger setpoint size can result in a larger deviation from the intended path, this is an important result that should be addressed when tuning the BLC.

B. Scenario 2: Different Local and Global Cost Function Weights

The second scenario is used to present a way to mitigate the BLC curvature effect that occurred in scenario 1. In order to achieve this result, the possibility and effect of different cost function weights are presented. The influence of minimizing the maximum control inputs (acceleration and heading change) by the local Bi-level Detailed Controller (BL-DC) instead of reducing the distance of the aircraft to the final destination is investigated. The controller parameters and scenario setup are summarized in Appendix A. The weights on the absolute maximum control inputs ($W_{d_{\alpha 2}}$ and $W_{d_{\beta 2}}$) are 10 and the weight on difference between consecutive control inputs ($W_{d_{\alpha 1}}$ and $W_{d_{\beta 1}}$) are both zero in this experiment. Moreover, it should be noted that the setpoints for the local Bi-level Detailed Controller (BL-DC) are determined at a higher rate of 100 seconds than the BL-RC sampling time of 300 seconds reducing N_{p_d} by a factor 3. Additionally, the maximum velocity boundaries for the BL-DC are bound between 180 m/s and 220 m/s. Instead of 180 and 200 m/s for the Single-level Accurate Controller (SL-AC) and the Bi-level Rough Controller (BL-RC).

Figure 12, portrays the trajectories of the SL-AC and the BLC, using the different cost function weights approach. It can be concluded that the unwanted curvature is no longer visible in the BLC trajectory. Instead, the path set out by the BLC follows that of the SL-RC while minimizing the required control input.

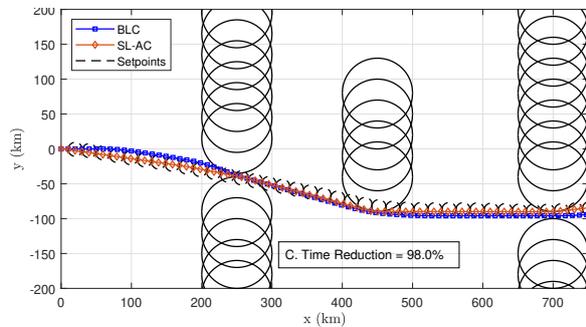


Fig. 12: Trajectory comparison of the BLC and the SL-AC for scenario 2.

When a single-level controller only uses W_f in its cost function it will opt to fly at the maximum allowed velocity since this is the optimal solution to decrease the distance between the aircraft and its destination. Therefore, the initial velocity of 200 m/s is also the maximum velocity constraint for the SL-AC and SL-RC in this scenario. The BL-DC is granted a wider velocity constraint and the maximum allowed velocity is set at 220 m/s. Even though the maximum velocity for the BLC is higher, the BLC does not opt for a maximum velocity solution, because its control objective is to minimize the control effort and not the distance to the destination. Furthermore, the local controller boundaries are wider than the SL-AC and BL-RC and the maximum heading change is 0.5 deg/s for the all controller versions, but 0.4 deg/s for the BL-RC. When looking at the BLC control inputs, in Figure 13, associated with the trajectory in Figure 12, it can be seen that there are only minor control inputs being applied by the BL-DC.

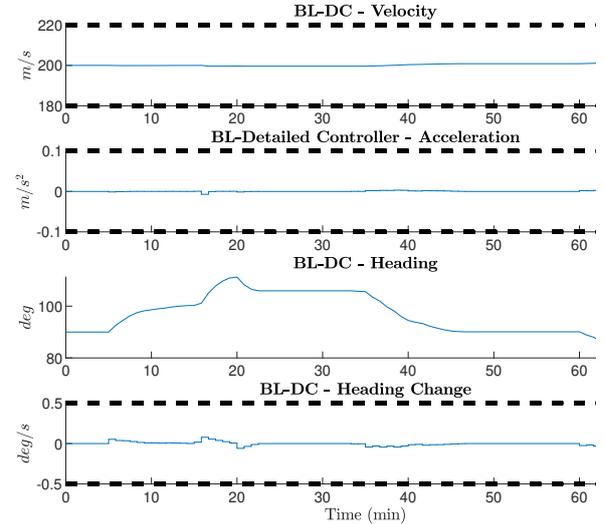


Fig. 13: Velocity, heading and optimized control inputs with $W_{d_f} = 0$ and $W_{d_{\alpha}}, W_{d_{\beta}} = 10$ for scenario 2

The heading and heading change controller input for the SL-AC can be found in Figure 14. Since the SL-AC already starts at the maximum velocity (200 m/s), the acceleration input remains zero.

Implementing a higher priority to W_{α} and W_{β} while using a single-level controller introduces a certain level of complexity and requires finding a balance between $W_{r_{\alpha}}$, $W_{r_{\beta}}$, and W_f . Additionally, because the distance to the destination decreases as time progresses, the cost function weight must change over time. If the cost function weights are static, the influence of $W_{r_{\alpha}}$ and $W_{r_{\beta}}$ for minimizing the cost function grows as the aircraft converges towards its destination.

This scenario is especially interesting, since the

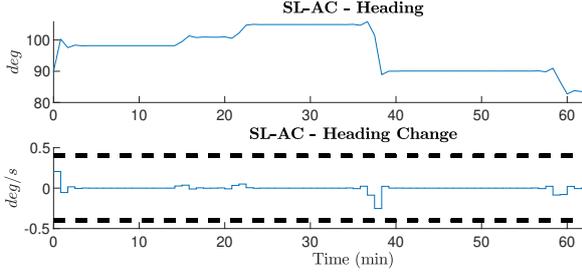


Fig. 14: Heading and heading change input SL-AC for scenario 2.

direct control layer of the BLC, does not take the final destination into account when determining its control inputs. However, it is capable of finding a smooth trajectory through the obstacle field. While reducing the total simulation time by 98%, compared to the baseline solution. It can thus be concluded that it can be beneficial to use different cost functions for both controller levels, to have each level focus on a different aspect of the solution. A figure showing the achieved level of detail by each of the controller methods as well as the corresponding SL-DC and SL-RC trajectories can be found in Appendix B.

The BLC trajectory (with $Wd_f = 0$) is comparable to the SL-AC trajectory while the control effort is lower. The different constraint boundaries for the velocity and the maximum heading change on both controller levels open up a possibility for different models and constraints on each controller level of the BLC.

Alternatively, when the BL-DC cost function is changed to place more emphasis on minimizing the distance to the destination ($Wd_f \neq 0$), a destabilizing effect starts to appear when the BL-DC velocity boundaries are wider than that of its leading rough controller. This effect can be seen in Figure 15, where the BLC oscillates between the setpoints.

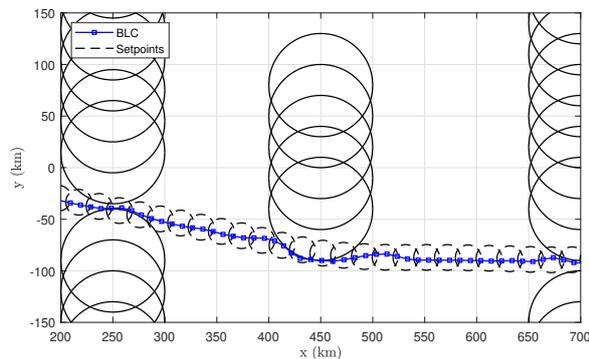


Fig. 15: Aircraft trajectory with with $Wd_f = 0.0001$ and $Wd_\alpha, Wd_\beta = 0$ for scenario 2

The control inputs shown in Figure 16 confirm the destabilizing control behavior. The aircraft accelerates

and decelerates and makes many heading changes to adhere to the setpoint constraints while also reducing its distance to the destination. Due to the different constraint setup, the BL-DC has the possibility of a higher velocity, creating an oscillating controller input.

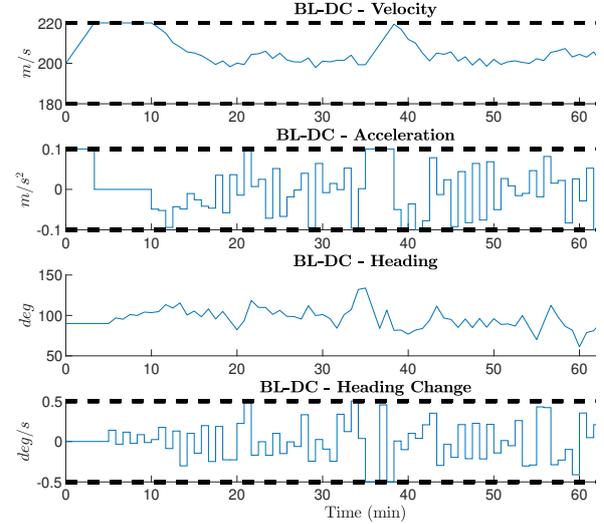


Fig. 16: Control inputs BL-DC for scenario 2 with $Wd_f = 0.0001$ and $Wd_\alpha, Wd_\beta = 0$.

When using different models or constraints on either level this effect must be addressed carefully, such that different models and constraints on one controller level do not interfere with the objectives of the other controller level.

C. Scenario 3: Measurement Updates

In all the simulations presented before, there has been no state measurement provided by the local Bi-level Detailed Controller (BL-DC) to the global Bi-level Rough Controller (BL-RC). The information flow has only gone from the global to the local controller. Scenario 3, presented in this section is used to explore the possibility of a 'two-way' information stream by using feedback from the BL-DC to the BL-RC. The starting points for each new global rough optimization are the states of the local detailed controllers, changing the hierarchical structure of the BLC by granting more autonomy to the BL-DC.

Several advantages but also some severe potential disadvantages should be considered. One advantage of providing a two-way information stream is that due to the faster control sampling rate of the BL-DC, it has more accurate state measurements, enabling better handling of fast disturbances. Furthermore, priority can be given to the BL-DC which can use different cost functions or more accurate models.

The main disadvantage is that a destabilizing effect can occur by providing feedback to the BL-RC. The BL-DC has the autonomy to adjust its heading

and velocity within the provided setpoint constraints. However, when the BL-DC operates at the boundaries of these setpoints, the global rough controller might have to make large control adjustments, that could even cause an infeasible solution.

The controller parameters and the scenario setup are presented in Appendix A. The setpoints for the BL-DC are determined at a higher rate of 150 seconds than the BL-RC sampling time of 300 seconds, reducing Np_d by a factor 2.

In Figure 17, two BLC trajectories are shown. Both versions incorporate a measurement update from the BL-DC to the BL-RC. However, each controller uses different cost function weights for the BL-DC. Where one BL-DC version uses Wd_f in its cost function, while the other presented BL-DC version has a weight on minimizing Wd_α and Wd_β . The weights on the absolute maximum control inputs ($Wd_{\alpha 2}$ and $Wd_{\beta 2}$) and the weight on the difference between consecutive control inputs ($Wd_{\alpha 1}$ and $Wd_{\beta 1}$) are all 10. The setpoint radius is 6.2 km and the maximum velocity is again equal for both controller levels.

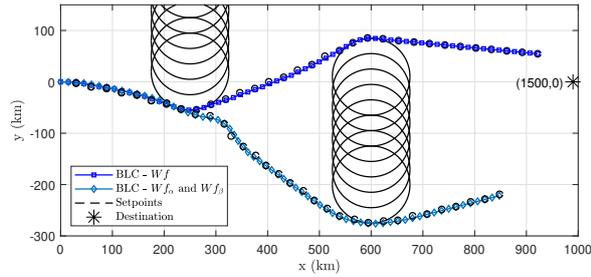


Fig. 17: Trajectory comparison of two BLC versions with different cost function weights using measurement updates, scenario 3.

As a result of the local perspective that the BL-DC has of the future, its optimal solution could deviate from the global controller's solution. Since the BL-DC can find a different controller input than the BL-RC, it can strafe away from the BL-RC setpoints, by as much as the setpoints constraints allow. This effect was already seen by the curvature in scenario 1 and again a slight curvature is visible in the Wd_f trajectory. If the local controller is given more authority by providing the starting positions to the BL-RC, the result can be unstable. Because the local controller can find control inputs within the allowed constraints, large correction by the BL-RC might be required to bring the aircraft back on the global optimal path.

It can be seen in Figure 17, that when using the BL-DC version with $Wd_f = 0$ and only Wd_α and Wd_β , the trajectory deviates from the optimal path. This can be explained by the fact that the BL-DC minimizes the control input, therefore its optimal solution is to minimize the heading change and stay at a constant ve-

locity. The solution of the local controller can deviate from what the global controller determines to be the shortest trajectory. Since the local controller provides the new initial conditions for the global controller, the BL-RC ends up on a sub-optimal trajectory.

Intermediate Dynamic Destinations

The possibility of utilizing the Bi-level Rough Controller (BL-RC) setpoints as intermediate destinations for the Bi-level Detailed Controller (BL-DC) is explored in this section.

In the simulations presented in Figure 18 the first two setpoints are again used to define the circular setpoint constraints. The difference is that instead of using the final destination in the cost function of the local BL-DC, the distance of the aircraft to the third setpoint is minimized. The BL-DC destination is, therefore, dynamically updated throughout the simulation.

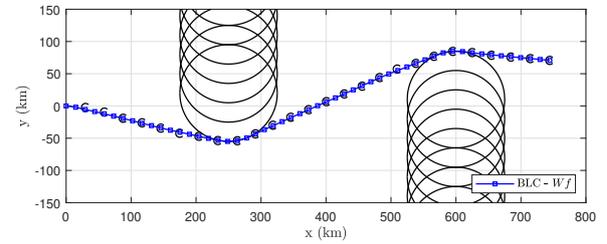


Fig. 18: Trajectory of the BLC with measurement updates and setpoint as destination with similar models on both levels.

The aircraft controlled by BLC in Figure 18, stays on the shortest path and the slight curvature that was present in Figure 17 is no longer visible. As expected, the BL-DC now follows the solution of the global BL-RC controller more closely. This is due to the global controller not only providing dynamic constraints, but also determining the reference (setpoints) for the local controller. The BL-DC effectively acts to minimize the error between its predicted states and the provided reference.

The BLC trajectory in Figure 18 is compared to the Single-level Accurate and Detailed Controller (SL-AC and SL-DC) trajectories with comparable controller parameters in Figure 19. It is clear that the SL-AC and BLC follow a similar path, while the SL-DC deviates from the shortest path until the forbidden area appears within its look-ahead window. Using this approach, the BLC trajectory approaches the baseline trajectory while improving the computational performance by more than 93%.

The level of detail of the BLC, SL-AC, and SL-RC can be seen in Figure 20. The shortcoming of the SL-RC is evident, while the level of detail is maintained using either of the other two control methods.

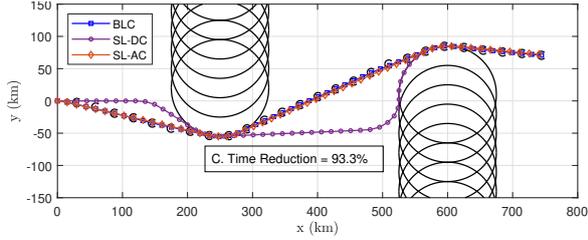


Fig. 19: Trajectory comparison of the BLC, SL-DC and SL-AC, all using similar models and controller parameters for scenario 3.

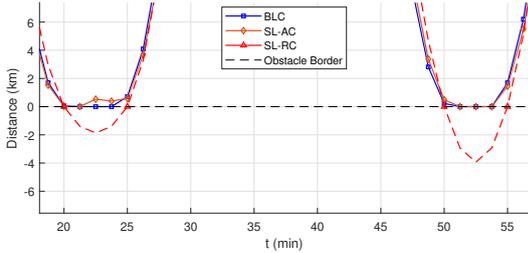


Fig. 20: Relative positional distance to the forbidden area boundary for scenario 3.

It can be concluded that when BL-RC also dictates the intermediate destinations to the BL-DC, the measurement updates do not necessarily destabilize the trajectory when the cost function weights are tuned appropriately.

Wider Local Constraints

It must be noted that in the simulation presented before, similar models and constraint boundaries have been used. This was done in order to mitigate the possibility that the initial velocity for the Bi-level Rough Controller (BL-RC) is already outside the feasible domain, when the BL-DC determines the initial states. In this section, it is shown what happens when the constraints on both levels start to contradict each other. In the simulation shown in Figure 21 the same scenario is presented, however a wider maximum velocity constraint (220 m/s) is implemented on the Bi-level Detailed Controller (BL-DC) level.

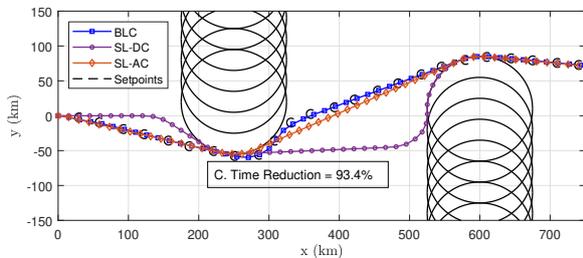


Fig. 21: Trajectory comparison of the BLC, SL-AC and SL-DC where the BLC is using wider constraints at the local controller level.

It can be seen in Figure 21, that the BLC still follows the baseline trajectory closely, even though there is a slight deviation from the optimal trajectory. However when taking a closer look at the control inputs from both levels of the BLC in Figure 22, some important conclusions can be drawn. The local BL-DC tries to fly at its maximum velocity since to minimize the distance between the aircraft and its destination. However, since the maximum velocity of the global controller is less than that of the local controller the two controllers have contradicting control inputs. The starting point of the optimization of the BL-RC is now already outside of its feasible solution space due to the feedback loop, and the global controller will, therefore, provide a negative acceleration input in order to stay within its velocity constraints.

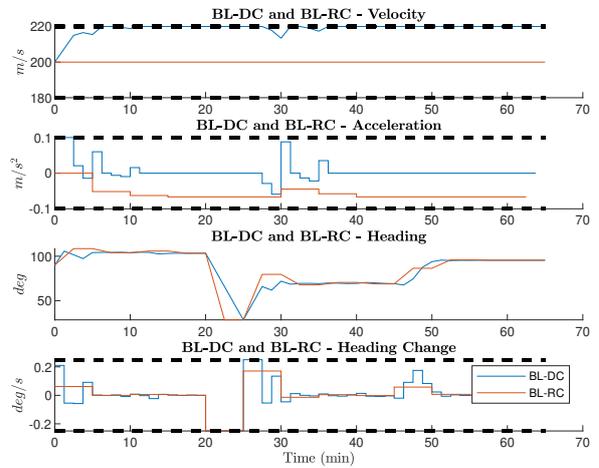


Fig. 22: States and controller input comparison for the BL-RC and BL-DC for scenario 3.

Therefore, the possibility of an infeasible optimization and contradicting solutions must be carefully considered when using a feedback loop from one controller to the other when the models and constraints are not exactly equal. It is clear that both controllers in this example have contradicting inputs.

Intermediate Destinations Without Measurement Updates

In this section, a final controller alternative is presented, in an attempt to fully remove the destabilizing behavior and to discuss the advantages that the Bi-level Controller (BLC) has over the Single-level Accurate Controller (SL-AC). This method uses no feedback from the local Bi-level Detailed Controller (BL-DC) to the global Bi-level Rough Controller (BL-RC), yet it does utilize the third setpoint as a destination for the BL-DC. Figure 23 shows a trajectory example generated using this approach with the cost function weights as shown in Table IV.

From the BLC cost function weights in Table IV, it

TABLE IV: BLC cost function weights for trajectory presented in Figure 23.

	W_α	W_β	W_f
BL-RC	0	0	0.0001
BL-DC	10	10	0.001

can be seen that the control input sequence is based on an optimization where minimizing the distance to the third setpoint is balanced with minimizing the control effort.

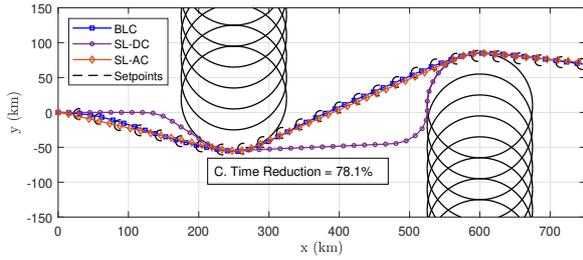


Fig. 23: Trajectory comparison without measurement updates and the third setpoint is used as a dynamic intermediate destination for scenario 3.

The velocity, velocity change, heading, and heading change are plotted in Figure 24. The BL-DC is allowed to increase its velocity up to 220 m/s, while the maximum velocity for the SL-AC is 200 m/s in this simulation. Considering the cost function setup of the SL-AC, which only uses W_f , the controller will have the aircraft fly at the maximum velocity for as long as possible, to reach the destination earlier. Therefore no fair comparison could be made if the controllers would use the same constraint boundaries.

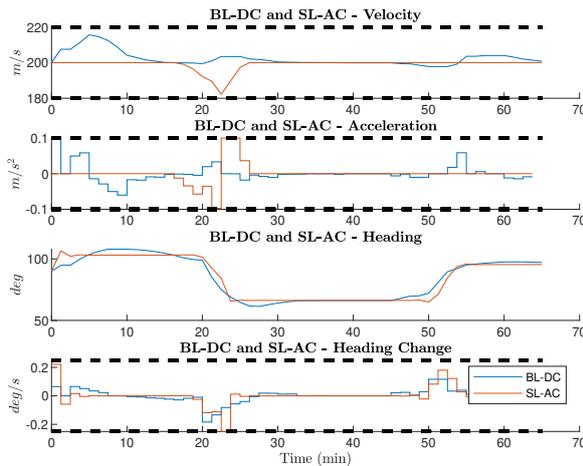


Fig. 24: States and controller input comparison for the SL-AC and BL-DC trajectories shown in Figure 23.

This setup results in three advantages. From Figure

24, the first important advantage of the current BLC emerges. A larger margin on the constraint boundaries of the detailed local control level can be allowed. This is effectively a possibility for implementing different models on both control levels. In this example, the maximum velocity boundary is wider for the local BL-DC than for the global BL-RC. It can be seen that the BL-DC uses this extra margin in the first 10 minutes of the simulation to increase its speed to fly around the forbidden area to reach its next setpoint in time. Since the BL-DC cannot take the same 'shortcut' through the forbidden area, it is forced onto a longer trajectory. However, it makes up for the extended trajectory by increasing its velocity. As a side effect, the BLC guides the aircraft faster through the obstacle field than when using the SL-AC. Because of the global BL-RC has a more efficient route than the SL-AC, due to the shortcuts. The BL-DC is capable to keep up with the BL-RC because it has the possibility to increase the velocity past the nominal speed of 200 m/s.

The second benefit of using this method is that there is no risk of the BL-DC deviating from the path determined by the BL-RC since there is no feedback. It was concluded from the results of scenario 3, that a destabilizing effect occurred due to the measurement updates provided by the BL-DC. For the version of the controller in this section, the hierarchy of determining the direction of the trajectory is granted to the global controller again.

Another gain of the current setup is that tuning the BL-DC is more convenient. The distance from the aircraft to the third setpoint is always within the same range, eliminating the need for dynamic cost function weights. These would be necessary when using a single-level controller that contains one or more cost function terms besides minimizing the distance to the destination.

When implementing different models or constraints on either level of the BLC, the constraints or models of the global controller have to be more conservative. It can be concluded from this example that the local controller can reach the global setpoint as a result of its less conservative model. The other way around might cause problems if the setpoint constraints cause a mismatch with different constraints than the BL-DC is subjected to. If the local controllers cannot reach the setpoint constraint due to a mismatch in models or model parameters, infeasible solutions might occur.

A downside of this setup is that there is no feedback from the local controller to the global controller. When disturbances and model imperfections start to play a more dominant role, the solutions of the local and global controller can accumulate over time. Future research could focus on implementing a method for some level of feedback from the local to the global

controllers, to make sure that the global controller and the detailed controller are able to provide feasible solution optimization solutions throughout the simulation. Feedback is increasingly important if disturbances are added and the scenarios become more complex. A possibility would be to decrease the feedback frequency or implement event-triggered feedback solutions. To reduce the risk of the local controller destabilizing the solution.

D. Scenario 4: Relative Computation Time

This section aims to quantify the relative computation time reduction that can be achieved when applying the Bi-level Controller (BLC) instead of the Single-level Accurate Controller (SL-AC) baseline. The method of using the third setpoint as a destination for the Bi-level Detailed Controller (BL-DC), without measurement updates, is used during the following simulations. The sampling times of the Bi-level Rough Controller (BL-RC) are the independent variables during this scenario. The look-ahead time of the BL-RC and SL-AC is fixed at 30 minutes, thus the prediction horizon of the SL-AC (N_{p_a}) is 60, according to Equation 27.

In Table V, the maximum and mean computation times for each iteration and the absolute computation time that was required to attain the trajectory in Figure 25 are listed. Since the control sampling time was set at 30 seconds, a maximum computation time of 42.4 seconds yields a problem in terms of real-time performance.

TABLE V: Maximum, mean and total computation times by the SL-AC for scenario 4.

Max (s)	Mean (s)	Total (s)
42.4	2.6	306.3

The detailed controller level of the BLC needs to attain the same level of detail as the baseline controller. Therefore the control sampling time of the BL-DC is also specified to be 30 seconds. A range of BL-RC sampling times between 60 seconds and 300 seconds are tested to evaluate the influences on the computation times.

The prediction horizon of the BL-DC (N_{p_d}) is related to the ratio of sampling times and the number of setpoints used ($N_{set} = 2$) by Equation 12. Furthermore, the look-ahead time of the BL-RC needs to remain constant for all simulations. Therefore, when varying the controller sampling rates, N_{p_r} can be determined using,

$$N_{p_r} = N_{p_a} \cdot \frac{c_d}{c_r} \quad (30)$$

Considering N_{p_a} is 60 and $c_d = 30s$, Table VI can be completed using Equation 30. Table VI shows the different combinations of BL-RC controller sampling times (c_r) and the corresponding prediction horizon for both controller levels that are tested.

TABLE VI: Rough controller sampling time (c_r) and corresponding prediction horizon (N_{p_r} and N_{p_d}) to attain the required look-ahead time of 30 minutes.

c_r	60s	90s	120s	150s	180s	300s
N_{p_r}	30	20	15	12	10	6
N_{p_d}	4	6	8	10	12	20

Figures 25 and 26, indicate two examples for different c_r values. It can be seen that the trajectory of the local Single-level Detailed Controller (SL-DC) changes due to the different sampling times of the global rough controllers. This is expected since the controller parameters of the BL-DC are dependent on the controller parameters of the BL-RC. When increasing k_r , k_d naturally increases due to the coupling via the setpoints constraints.

In Figure 25, for $k_r = 90$ seconds, the look-ahead time of the BL-DC is 3 minutes, which is too short for the SL-DC to find a feasible trajectory.

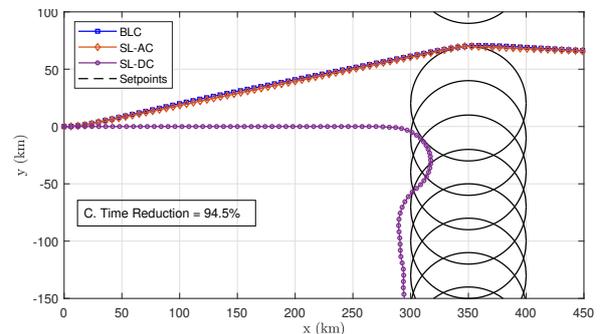


Fig. 25: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 90s$, scenario 4.

In Figure 26, the simulation with $k_r = 300$ seconds is shown. It is apparent that when the sampling time of the BL-RC decreases, the SL-DC starts to approach the optimal trajectory. In this simulation, the look-ahead time of the local controller is 10 minutes and because of the larger prediction horizon, the SL-DC starts to approach the baseline solution of the SL-AC. The other trajectories, corresponding to the other global sampling times, are presented in Appendix C.

The first conclusion that can be drawn from Figure 27, is that the computational efficiency initially grows for an increasing c_r . This effect can be contributed to a decreasing prediction horizon on the rough control level since the look-ahead time is fixed. Nevertheless,

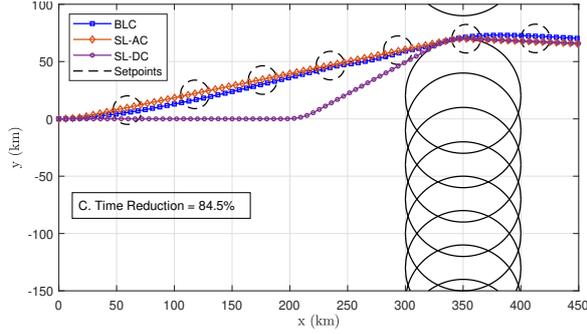


Fig. 26: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 300s$, scenario 4.

a decrease in the computational efficiency is observed when c_r increases past 120 seconds, for this scenario and controller setup. This can be explained by the fact that a larger Np_d is required when increasing the BL-RC controller sampling time. Consequently, the computation time for each optimization iteration of the BL-DC required more computational effort.

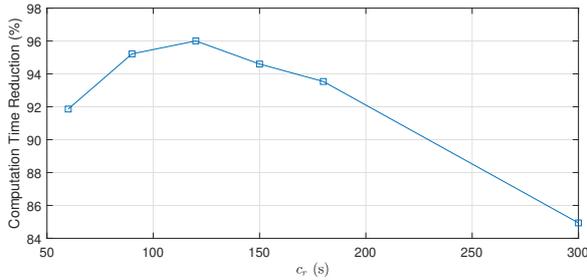


Fig. 27: Computation time reduction using the BLC compared to the SL-AC with a constant c_d , scenario 4.

The difference in computation times of both control levels can be seen in Figure 28, where the mean computation times for both controller levels are presented.

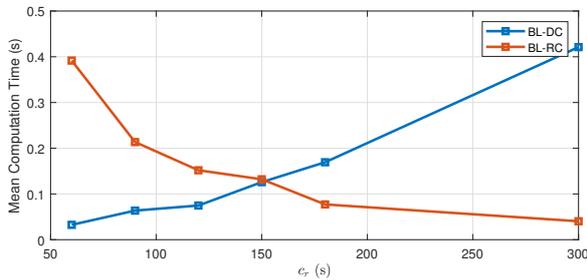


Fig. 28: Mean optimization computation times during entire simulation for each of the BLC control levels, scenario 4.

Finally, it should be noted from Figure 29 that the maximum computation times are also influenced by the size of c_r and that it follows a similar trend.

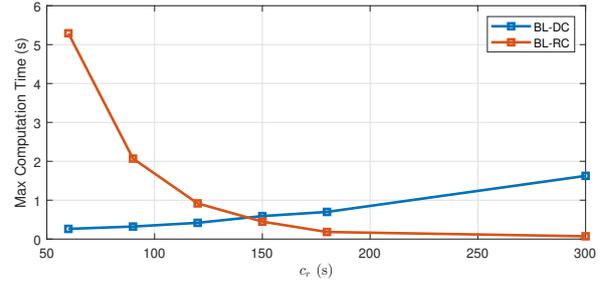


Fig. 29: Maximum optimization computation times during entire simulation for each of the BLC control levels, scenario 4.

It is possible to reduce the computation times of the BLC even further, by using intermediate setpoints. When c_s is smaller than c_r the setpoints are placed closer together. This will allow for a reduction of Np_d , while it has no effect on Np_a . The resulting relative computation times can be seen in Figure 30, where the same SL-AC simulations are compared to a BLC with $c_s = c_r$ and with $c_s = 0.5 \cdot c_r$. The corresponding trajectories with intermediate setpoints can be found in Appendix C.

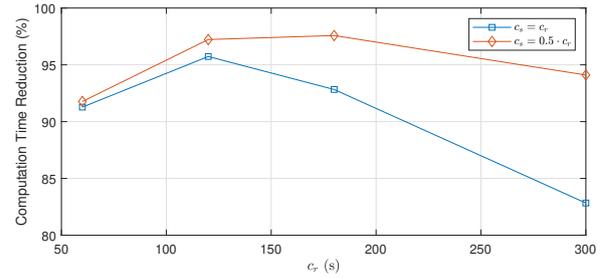


Fig. 30: Computation time reduction BLC compared to SL-AC with intermediate setpoints.

The computational efficiency gained by the BLC compared to the SL-AC depends on the relative control sampling times and the prediction horizons of both controller levels. When c_d is fixed, based on the required level of detail of the system and the ratio of c_s/c_d grows (Where $c_s \leq c_r$), the prediction horizon of the BL-DC must increase due to coupling by the setpoint constraints (see Eq. 12 and Eq. 13). The mean and max computation times of the BL-DC also increase for a larger Np_d , compromising the real-time performance.

The ratio of controller sampling times also have a relative performance influence besides the already mentioned computation time effects. The increased look-ahead time of the BL-DC, which is required when c_r increases, would reduce the performance benefits between the SL-AC and SL-DC. Due to the larger look-ahead time of the detailed controller, it becomes increasingly capable of controlling the aircraft

without the guidance of the BL-RC. Therefore, when increasing c_r , one must be aware that the local prediction horizons will increase for this controller setup. Increasing the computational effort and performance of the single-level detailed controllers, mitigating the added benefit of the bi-level controller structure. The introducing of intermediate setpoints partly mitigates this effect, but requires more research to find a systematic implementation approach.

In contrast to the previous effect, the opposite is also true. In case c_r is too small and approaches the controller sampling rate of the local controller, the level of detail that is achieved by the rough bi-level controller approaches that of the local controller level. Reducing the additional performance gain in terms of accuracy from the addition of the local controller level.

Nevertheless, it is evident that the usefulness of applying two control levels, in terms of the computational efficiency and accuracy, increases as the difference between the sampling times increases. However, in order to get the highest potential out of the bi-level controller structure, (online) tuning of the controller sampling times is essential and would be a valuable addition for further research.

VII. SIMULATION RESULTS: MULTI AIRCRAFT SIMULATIONS

In this section, the second part of the research and exploration of the bi-level MPC structure is presented. In the previous sections, the difference in look-ahead times and control sampling times of both controller levels was utilized to control one aircraft. The effect of different sampling times and look-ahead time of both control levels was explored and the benefits and drawbacks were presented. The second part, presented in the following sections entails the spatial distribution of both control levels, where the control task is also separated based on the amount of aircraft controlled by each controller. This creates a division of centralized and decentralized control, where the global BL-RC operates as a centralized controller and the local BL-DC are multiple separate decentralized controllers.

A. Scenario 5 and 6: Multi Aircraft Scenarios

For the fifth scenario, another aircraft is added to the simulation, justifying the terms, centralized and decentralized control. The global rough controller acts as a centralized controller and computes setpoint constraints for both aircraft. The local detailed controllers operate in a decentralized manner and have no knowledge of each other, they are only following the centralized setpoint instructions. The controller parameters can be found in Appendix A.

In the simulation results presented in Figure 31, it can be seen that the centralized controller has successfully guided one aircraft around the forbidden areas

while guiding the other aircraft through the forbidden areas. The problem posed is exactly symmetrical numerically. Both aircraft fly at equal velocities and start at equal but opposite heading angles. They are positioned at equal lengths from the forbidden areas and have a final destination at $x = \pm\infty$ and $y = 0$ km. Furthermore, both aircraft use the same cost function weights, constraint boundaries, prediction horizons, and sampling times. Since the spacing between both forbidden areas is only 2 km, there is no room for the aircraft to pass each other (the aircraft separation requirement is 5 nautical miles). One aircraft, therefore, has to deviate from the shortest path, or a conflict is imminent. Only the global controller could determine that solution, the local Single-level Detailed Controllers (SL-DC) without knowledge of the other aircraft would not have been able to prevent this collision on its own.

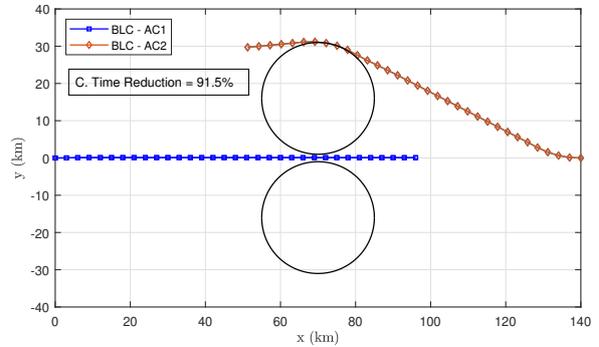


Fig. 31: Trajectories of two aircraft around forbidden areas using the BLC, scenario 5.

The Single-level Accurate Controller (SL-AC) trajectory presented in Figure 32. This trajectory is nearly identical to that of the BLC, yet the BLC finds the same trajectory more than 90% faster.

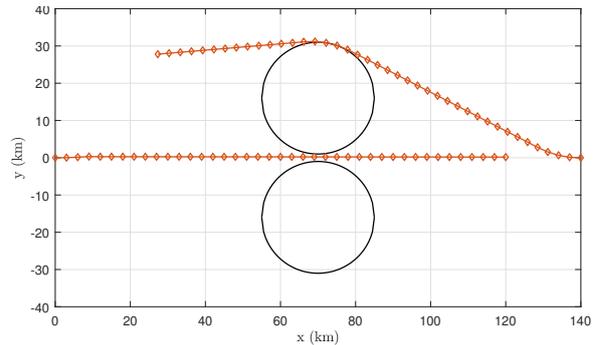


Fig. 32: Trajectories of two aircraft around forbidden areas using the SL-AC, scenario 5.

The real-time performance of the controllers can be evaluated using the computation times for each iteration in Figure 33. It can be seen that the maximum computation time of the SL-AC is more than 5

times longer than the computation times for the Bi-level Detailed Controller (BL-DC), to complete each optimization. Furthermore, the local BL-DC is 5 to 10 times faster than the global Bi-level Rough Controller (BL-RC) level.

The sampling time of the local detailed controllers is 15 seconds in this scenario. The maximum computation times required by the SL-AC are over 50 seconds. Whereas, the BL-DC found a control input within 1 second and the BL-RC took less than 10 seconds to find the optimal solution every new iteration.

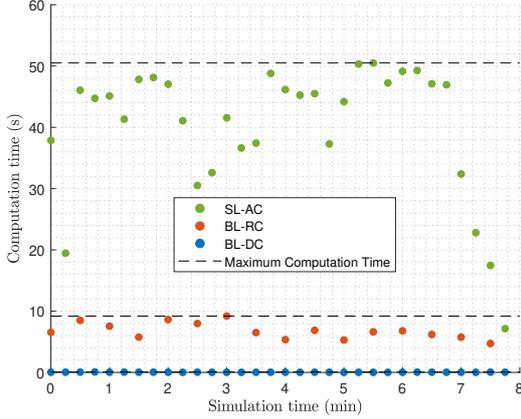


Fig. 33: Computation times of the BLC and the SL-AC for every iteration of two aircraft simulation shown in Figures 31 and 32.

The two aircraft simulation can be extended to multiple aircraft. Figure 34 illustrates an example with four aircraft flying around a circular forbidden area. The starting situation is symmetrical around the x and y -axis. The rough centralized controller is capable of determining conflict-free trajectories, by choosing a side for each controller. Again only the BL-RC with global knowledge of the system could determine such a solution by determining conflict free trajectories for each aircraft. The computational effort is reduced by 98% when using the BLC instead of the SL-AC methods for the same scenario.

The multi aircraft simulations presented in this section show the potential benefits of having a spatial separation on both control levels. The rough long-term centralized controller with global knowledge of the system can quickly guide the aircraft in a general conflict free direction, while the detailed decentralized controllers optimize that trajectory on a local scale. When using multiple aircraft the computational efficiency becomes even more important. With every aircraft added to the system the computation time increases quadratic, due to the full coupling of all aircraft by the separation constraint.

The BL-RC and the SL-AC are subjected to the separation constraint for their optimization, increas-

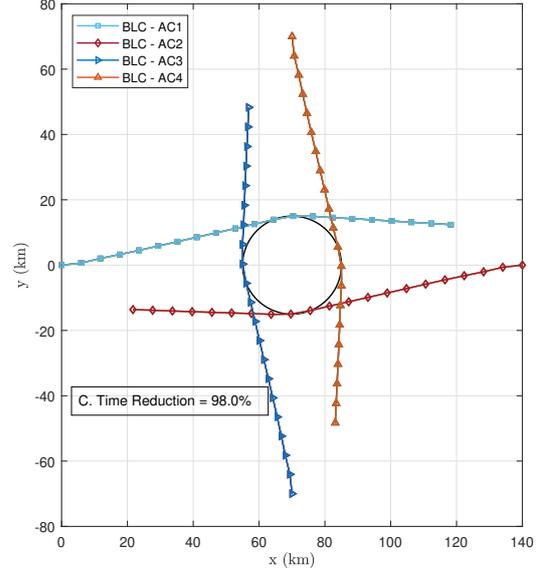


Fig. 34: Simulation with four aircraft that need to avoid each other and the central forbidden area.

ing the computational burden for each aircraft added to the system. Due to the longer sampling time of the BL-RC, it has fewer decision variables and the optimization can be executed more efficiently. Since the BL-DC controllers only control one aircraft, the number of aircraft does not influence the computation time of the BL-DC level and it is negligible compared to the BL-RC. It should be noted that since no parallel computation is used for these scenarios the BL-DC optimization is performed sequentially. Therefore, when implementing a parallel computation technique, which the BL-DC is well suited for, the computational effort reduction that can be achieved could be even higher.

B. Scenario 7: Conflict Resolution Without Forbidden Areas

This section serves to indicate a potential risk when the BLC is used for aircraft separation. Two main problems need to be addressed when applying the BLC for aircraft separation. The first problem is the discretization. Since the minimum separation requirement (δ) is evaluated at discrete moments in time, a loss of separation can happen in between two time evaluation moments. A scenario where a potential conflict can occur is schematically presented in Figure 35. It is evident that when setpoints are too far apart, conflicts can occur in between two evaluation moments.

When two aircraft fly towards each other in a head-on conflict scenario, the maximum discretization period is given by,

$$c \leq \frac{\delta}{2 \cdot v} \quad (31)$$

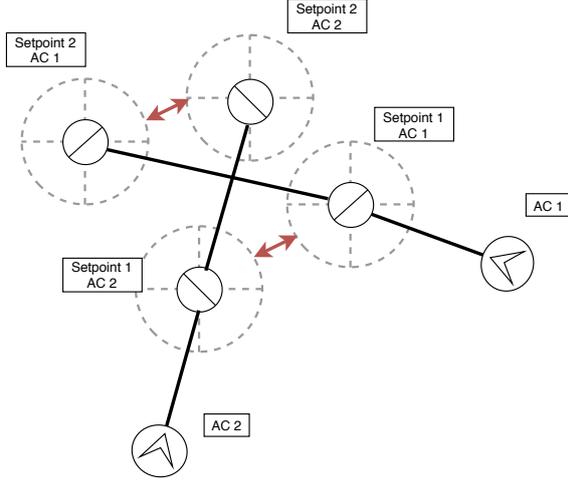


Fig. 35: Potential conflicts in between setpoints as a result of discretization and large sampling times.

For example, if the minimum aircraft separation is 5NM (9.26 km) and two aircraft are flying towards each other at a head-on heading with a velocity of 200 m/s, loss of separation can occur when the sampling time (c) is larger than 23 seconds. Therefore a relatively short control sampling time is required for the controller that incorporates the separation constraints, the global Bi-level Rough Controller (BL-RC) in this case.

The second issue that needs careful consideration is that the decentralized local detailed controllers do not have a separation constraint implemented. The controller algorithm is based on the global rough controller to provide a solution space for the local detailed controller based on setpoints. However, inside the solution space there can still be infeasible solutions, as is indicated by the green area in Figure 6. For the area avoidance constraint, this did not impose any problems, since the obstacle avoidance constraint is also part of the detailed control loop also. The separation constraint is not added to the local controller level, thus the setpoint solution space must be safe from an aircraft separation point of view.

However, for a multi aircraft scenario, the actual separation can become less than the separation minimum used by the separation constraint. Two separation distances need to be evaluated when using the current setup. Using Figure 36, the actual aircraft separation can be derived.

The minimum separation constraints evaluates different aircraft positions at the same moment in time. Taking the example presented in Figure 36, it can be seen that in between setpoints 1 and 2, the two aircraft will eventually be closer together, than the minimum separation it was evaluated for. The actual separation as a result of a single-level controller (δ_r)

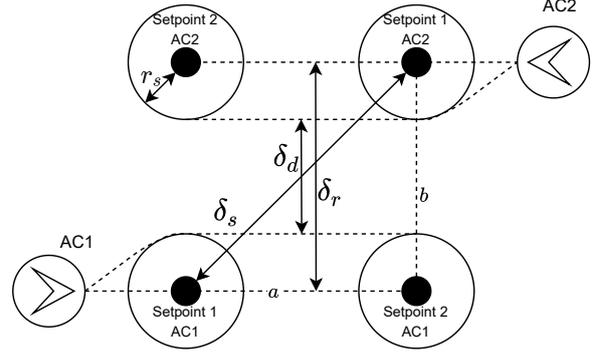


Fig. 36: Differences and causes of the mismatch between actual and evaluated aircraft separation for a head on conflict, where $a = v \cdot c$ and $b = \delta_r$.

should therefore be evaluated using,

$$\delta_r = \sqrt{\delta_s^2 - (v \cdot c_r)^2} \quad (32)$$

The local Bi-level Detailed Controller (BL-DC) has no separation constraint and is free to move anywhere within the setpoint area constraint. The actual aircraft separation, as a result of the detailed control level (δ_d), can reach as low as,

$$\delta_d = \delta_r - 2 \cdot r_s \quad (33)$$

In the worst case scenario, two aircraft are at the limits of the setpoint constraint radius (r_s). Therefore, in order to achieve the desired separation distance on the separation constraint an extra margin is required and should be computed using,

$$\delta_s = \sqrt{(\delta_r + 2r_s)^2 + (v \cdot c_r)^2} \quad (34)$$

Figure 37 shows a simulated version of the previously explained concept. Two aircraft are flying towards each other, creating a head-on conflict. Each aircraft starting and final positions are on the y-axis, therefore each aircraft opts to stay as close to the y-axis for the most efficient route. It can be seen that both aircraft trajectories fly at their respective setpoint boundaries.

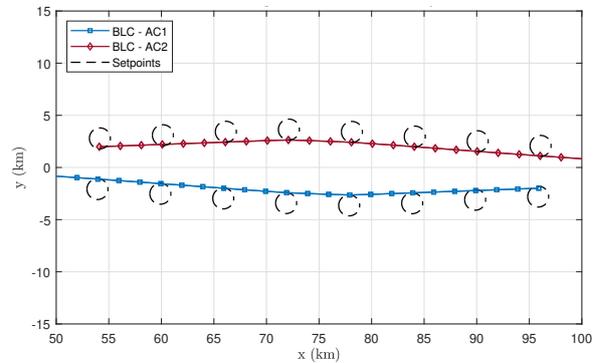


Fig. 37: Simulation of two aircraft flying at a head-on trajectory without central forbidden area, scenario 7.

Consequently, the required separation of 5NM is violated if this effect is not designed for, as can be seen in Figure 38.

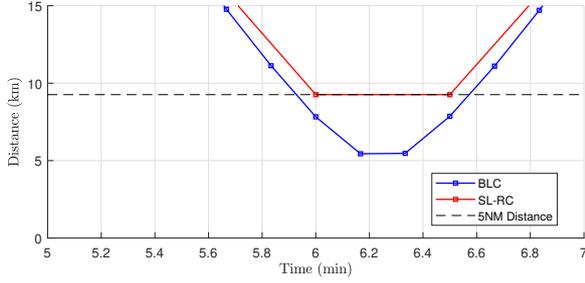


Fig. 38: Relative distance between two aircraft flying at a head-on conflict. The aircraft separation requirement is violated by the BLC due to the allowed setpoint region.

To obtain an adequate separation margin, Equation 34 can be applied, which has a consequence that the airspace is used less efficiently by using larger margins. Alternatively, the control sampling times can be decreased, reducing the roughness of the discretization at the cost of a higher computational burden. Or a separation constraint could be implemented on the lower control level. The latter option is discussed in the next section.

C. Scenario 8: BL-DC Separation Constraint

This section serves to indicate how adding a separation constraint to the local Bi-level Detailed Controller (BL-DC) can be implemented to attain an adequate level of separation between aircraft. In Figure 39, first the baseline trajectory using the Single-level Accurate Controller is shown (SL-AC).

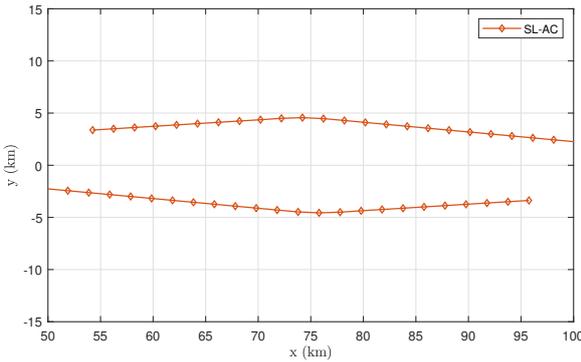


Fig. 39: Simulation of two aircraft flying at a head-on trajectory without central forbidden area using the SL-AC, scenario 7 and 8.

In the simulation using the BLC, a change has to be made to the BLC concept as well as, the controller setup. As was explained using the algorithm presented in section IV, the global rough controller and the

local detailed controller operate in a different way. To create a centralized and decentralized version of the controllers, the global rough controller finds a global optimum for all aircraft in the system simultaneously, mitigating the sequential aircraft separation issues. Whereas the local detailed controller acts as a decentralized controller and the aircraft trajectories are computed sequentially, such that the local detailed controller could potentially be separated over different computers. Since the aircraft are not coupled on the local detailed controller level the sequential optimization does not affect the results. To make sure adequate separation of 9.26 km is maintained by both aircraft, the separation constraint for this scenario is now implemented on both controller levels. Figure 40 shows the trajectory of two aircraft with this change implemented.

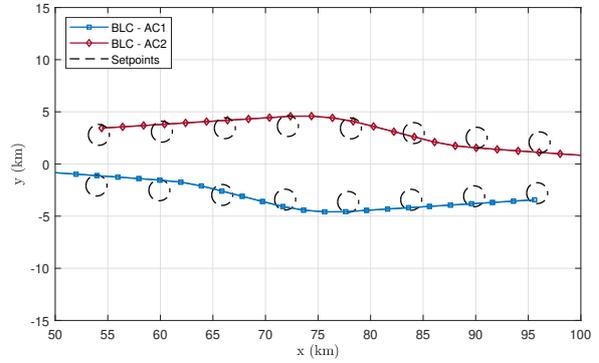


Fig. 40: Simulation of two aircraft flying at a head-on trajectory without central forbidden area using the BLC with the separation constraint on both controller levels, scenario 8.

When a separation constraint is applied on the detailed controller level also, the aircraft are no longer decoupled on this level and the sequential optimization would affect the result. The sequential optimization is removed and the local detailed controller uses the same simultaneous optimization technique that the centralized controller uses. The aircraft separation is presented in Figure 41 where it can be confirmed that both the BLC and the SL-AC maintain the required aircraft separation, using the new setup. Even though it can be observed that the BLC trajectory (Fig. 40) is sub-optimal compared to the SL-AC trajectory (Fig. 39).

Despite the fact that the local vs. global optimization strategy is maintained and the setpoint can still be used to guide aircraft in the most efficient direction, the centralized vs. decentralized division is lost in this setup. In order to maintain the decentralized control characteristics on the detailed control level, while maintaining the aircraft separation properties, more research must be conducted.

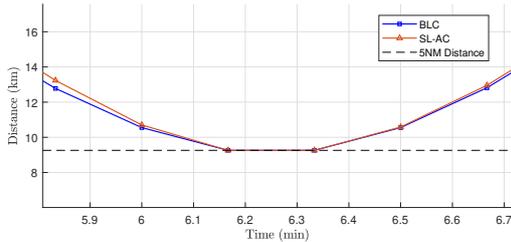


Fig. 41: Relative separation distance between two aircraft flying at a head on trajectory using either the BLC or the SL-AC, corresponding to Figures (39-40)

VIII. DISCUSSION

The difference between the global Bi-level Rough Controller (BL-RC) and the local Bi-level Detailed Controller (BL-DC) is not limited to the contrast in the prediction horizons and the control sampling times. The difference is also defined by the centralized and decentralized characteristics of the control levels. Besides the mentioned benefits resulting from the computational advantages with a minimal compromise in the level of detail and controller performance, other advantages can be achieved by using the potential of multiple control levels, as has been shown by the multi-agent simulations in the previous sections.

If different models are applied on either level, the centralized controller typically has less accurate knowledge of the model parameters (weight, aerodynamic characteristics, etc.). This could be because of the lack of information or when these parameters contain sensitive information that can not be shared, an example is given in [7]. The models used by the decentralized controllers can be more accurate, which enables the decentralized controller to perform a more accurate optimization. This could be a reason to apply a bi-level control structure as proposed in this study. The setpoint constraints allow for the local detailed controller to deviate from the rough control solutions, within a safe margin. The case study in Section VI-B shows that wider constraint margins can be applied to the lower level controller, which enables different models to be applied on both levels. Even though careful considerations must go into aligning the two controllers when different models are applied and feedback is desired.

The second possibility that is opened up by the BLC structure, is to handle disturbances more accurately. The rough centralized controller typically has less knowledge of the fast disturbance in comparison to the decentralized controllers. The decentralized local detailed controller could adequately respond to fast disturbances due to its higher control sampling time and possibly more accurate measurements. While the slow centralized controller can adequately respond to slow disturbances than span longer time periods, an

entire area of high turbulence for example. This would create an analogy with the forbidden areas used in this thesis.

Another opportunity that emerges due to the hierarchical control structure is that each aircraft can optimize a cost function with different weights, as was shown in this thesis. This creates a parallel with a realistic scenario where every airline can determine their preferred cost index, based on their business model. The cost index that a certain aircraft operates at does not have to be known by, or communicated to, the centralized controller. This allows for a different balance between flight time and fuel flow for each aircraft for example.

Decentralized multi-agent separation is an often researched subject in literature [7][24][12][9]. An optimization sequence is not explicitly required when a system is fully decentralized and agents do not rely on each other's decisions. However, for the highly coupled air traffic control field where safety is the priority, some level of cooperation is required. Therefore, when using a bi-level control structure, the global optimization and safety constraints can be moved to another hierarchical control level. In such a system, the individual aircraft do not have to interact with each other, since the centralized controller handles the interaction between the systems in terms of separation. The decentralized controlled aircraft transmit their state information to a centralized controller, which combines the data to perform a global optimization. This is much like the traditional way in which air traffic is managed. Where a single human controller manages multiple aircraft to guarantee the safety constraints are adhered to.

The Bi-level model predictive control structure opens up several conceptual advantages besides the improvements in computational efficiency. The potential of the BLC system could be explored, possibly also within different fields of engineering, such as drones or small robots.

IX. CONCLUSION AND RECOMMENDATIONS

This study aimed to investigate the integrating link between a global centralized MPC and one or more local decentralized MPC levels with different prediction horizons and different control sampling times. A clear synergy can be attained where the combined bi-level controller performs better than the sum of its parts for an air traffic control application. Even though many important aspects need to be considered when applying multiple model predictive control levels in a bi-level controller structure.

On a global scale, the Bi-level Controller (BLC) is capable of finding a more optimal path in terms of minimal flight distance than a single-level local

detailed controller. The long term guidance of the Bi-Level Rough Controller (BL-RC) can aid the local Bi-Level Detailed Controller (BL-DC) into the general solution space, by providing setpoint constraints. Nonetheless, on a local scale, the BLC operates at the faster controller sampling rate of the local detailed controller, resulting in a more accurate control solution than that of the global rough controller. Therefore, accuracy is gained at the short-term perspective enabling a more detailed control solution in terms of forbidden area avoidance. Due to the receding horizon properties of MPC the short term horizon with the required level of detailed advances as time progresses.

The bi-level controller has been proven to be capable of cooperatively guiding an aircraft past forbidden areas towards their destinations. Furthermore, the developed control system has the potential to be extended for multi-agent systems, where it can aid in reducing the computational complexity. It can be concluded that the baseline desired performance can be approached with the BLC with increased computational efficiency. It can even be argued that the BLC has preferable characteristics compared to the single-level accurate baseline controller because it can be easier to tune the cost function weights, especially when applying different cost function weights to either control level of the BLC.

Due to the lower long term predictability of global air traffic and the accumulation of model uncertainties and disturbances in long term prediction, the relevance of far ahead prediction can be questionable. Therefore, by only using far ahead prediction for general guidance a compromise in the long term accuracy of the control solution is acceptable. Due to the receding horizon properties of MPC, providing only adequate short term detail does not appear to have a negative outcome when comparing the complete trajectories of the BLC to the single-level baseline MPC.

Furthermore, it has been shown that it is possible to implement different models, model parameters, constraints, or constraint margins on either level of the BLC. However, the higher level controller must provide feasible setpoint regions for the lower level controller. Therefore, the BL-RC must operate with more conservative constraint boundaries or the setpoints constraints might yield an infeasible optimization on the local controller level.

It must be noted that when applying measurement feedback from the BL-DC to the BL-RC, a risk of destabilization appears. The control inputs of the BL-DC and BL-RC can vary largely. This can lead to situations where the control input of the local controller can drift the system away from the intentions of the BL-RC for as far as the setpoints allow, resulting in a sub-optimal solution. If both controller have different models, constraints, or cost functions, the

control inputs might yield contradicting commands. If the local controllers have less conservative constraints and provide updates to the global rough controller, the initial states might fall outside the feasible region of the global controller.

Future research could focus on reducing the destabilizing behavior and improving the possibility for the implementation of feedback loops from the local to the global controller. One option could be to reduce the feedback frequency or apply event triggered feedback.

Furthermore, the addition of disturbances in the form of a wind field could provide valuable insights into the benefits of a faster control sampling rate on the local controller levels. It would also further highlight the need for a feedback loop and measurement updates.

Moreover, running real-time simulations where the computational delays have a significant effect on the control performance might underline the performance of the BLC over the baseline controller. The simulations in this thesis have not been performed in real time, and the simulation was paused every time a new control input sequence needed to be optimized. Measurement delays have, therefore, not had any effect on the results even though they can be a significant factor. Especially when the optimization computation time exceeds or approaches the control sampling rate, as was often the case when using the single-level accurate controller.

The tuning parameters of the bi-level controller are case specific. The sizes of the forbidden areas, the complexity of the required path, the distance between forbidden areas, the distance between the aircraft, require different controller tuning parameters. To achieve the desired result, more research into the online tuning of the controller parameters can contribute to the operational performance of the controller.

The different controller sampling times of the BLC have been a point of focus during this thesis and is one of the main aspects that differentiate the distinct controller levels. It has been shown that when the global rough control sampling time is relatively small and approaches the detailed controllers' control sampling time, the added benefit of the local controller is reduced and the total computation time increases. Alternatively, when the controller sampling time of the rough controller becomes too large, the prediction horizon of the detailed controller needs to increase (when the level of detail is maintained). The increased look-ahead time on the local level usually results in an increased controller performance in terms of finding the optimal path. This makes the single level detailed controller increasingly capable of operating autonomously. Assuming that the local controller sampling time is a fixed requirement, tuning the rough controller sampling time is of vital importance. In

order to achieve the optimal potential from the bi-level control structure, future research in the (online) tuning of the rough controller sampling rates is highly recommended.

When intermediate setpoints are applied an extra tuning parameter is added to the controller, the setpoint sampling time. The setpoint sampling times, in this project, have been tuned for each specific scenario separately and remained constant during each scenario. Much potential for improving the performance and computation time can be gained by extending research towards using intermediate setpoints and tuning the setpoint sampling time. This could be done online such that it can be changed dynamically.

Optimizing the computational efficiency of the bi-level controller to achieve real-time control is desirable but was not a focal point of this research project. However, it has been shown that the BLC control structure can contribute to achieving real-time performance. Furthermore, the parallel computation of the centralized and decentralized control levels has not been part of this thesis and could be valuable additions to further increase computational efficiency.

Additionally, proving stability, robustness, and convexity of the solution have not been addressed explicitly. These are important topics for future research.

Finally, the possibility to use different models and model parameters on either of the two controller levels can be a valuable addition for future research. Using more accurate complex models on the detailed control level would further highlight the potential of the combined centralized and decentralized bi-level controller.

REFERENCES

- [1] Eurocontrol. European Aviation in 2040. Challenges of Growth. page 40, 2018.
- [2] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovskiy. The development of model predictive control in automotive industry: A survey. In *2012 IEEE International Conference on Control Applications*, pages 295–302, Dubrovnik, Croatia, 2012.
- [3] H. Zheng, J. Wu, W. Wu, and R. R. Negenborn. Cooperative distributed predictive control for collision-free vehicle platoons. *IET Intelligent Transport Systems*, 13(5):816–824, 2019.
- [4] M. Nolte, M. Rose, T. Stolte, and M. Maurer. Model predictive control based trajectory generation for autonomous vehicles — an architectural approach. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 798–805, Los Angeles, CA, USA, 2017.
- [5] M A Brdys, M Grochowski, T Gminski, K Konarczak, and M Drewa. Hierarchical Predictive Control of Integrated Wastewater Treatment Systems. *Control Engineering Practice*, 16:751–767, 2008.
- [6] Maciej Lawrynczuk, Piotr Marusak, and Piotr Tatjewski. Multilayer and Integrated Structures for Predictive Control and Economic Optimization. *IFAC Proceedings Volumes*, 40:198–205, 2007.
- [7] L. Ferranti, R. R. Negenborn, T. Keviczky, and J. Alonso-Mora. Coordination of multiple vessels via distributed nonlinear model predictive control. In *2018 European Control Conference (ECC)*, pages 2523–2528, Limassol, Cyprus, 2018.
- [8] Cyril Allignol, Nicolas Barnier, Nicolas Durand, and Jean-Marc Alliot. A new framework for solving en route conflicts. *Proceedings of the 10th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2013*, 21, 06 2013.
- [9] G. Chaloulos, P. Hokayem, and J. Lygeros. Distributed hierarchical mpc for conflict resolution in air traffic control. In *Proceedings of the 2010 American Control Conference*, pages 3945–3950, Baltimore, MD, USA, 2010.
- [10] Lesley A. Weitz and Xiaoli Bai. Using Model Predictive Control for Trajectory Optimization and to Meet Spacing Objectives. *AIAA Guidance, Navigation, and Control Conference, 2018*, 2018.
- [11] Xin Min Tang, Ping Chen, and Bo Li. Optimal Air Route Flight Conflict Resolution Based on Receding Horizon Control. *Aerospace Science and Technology*, 50:77–87, 2016.
- [12] Kouamana Bousson. Model predictive control approach to global air collision avoidance. *Aircraft Engineering and Aerospace Technology*, 80:605–612, 10 2008.
- [13] Egidio D’Amato, Massimiliano Mattei, and Immacolata Notaro. Distributed reactive model predictive control for collision avoidance of unmanned aerial vehicles in civil airspace. *Journal of Intelligent & Robotic Systems*, 06 2019.
- [14] Egidio D’Amato, Massimiliano Mattei, and Immacolata Notaro. Bi-level Flight Path Planning of UAV Formations with Collision Avoidance. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 93(1-2):193–211, 2019.
- [15] Yoshiaki Kuwata and Jonathan P. How. Cooperative Distributed Robust Trajectory Optimization Using Receding Horizon MILP. *IEEE Transactions on Control Systems Technology*, 19(2):423–431, 2011.
- [16] Carlos E. Garcia, David M. Prett, and Manfred Morari. Model Predictive Control : Theory and Practice a Survey. *Automatica*, 25(3):335–338, 1989.

- [17] Walid Al-Gherwi, Hector Budman, and Ali Elkamel. Robust Distributed Model Predictive Control: A Review and Recent Developments. *Canadian Journal of Chemical Engineering*, 89(5):1176–1190, 2011.
- [18] Fatemeh Asadi and Arthur Richards. Ad Hoc Distributed Model Predictive Control of Air Traffic Management. *IFAC-PapersOnLine*, 28(25):68–73, 2015.
- [19] Piotr Tatjewski. Advanced Control and On-line Process Optimization in Multilayer Structures. *Annual Reviews in Control*, 32(July 2007):71–85, 2008.
- [20] Veronica Adetola and Martin Guay. Integration of Real-time Optimization & and Model Predictive Control. *Journal of Process Control*, 20, 2010.
- [21] Riccardo Scattolini. Architectures for Distributed and Hierarchical Model Predictive Control - A review. *Journal of Process Control*, 19(5):723–731, 2009.
- [22] George J. Pappas, Gerardo Lafferriere, and Shankar Sastry. Hierarchically Consistent Control Systems. *IEEE Transactions on Automatic Control*, 45(6):1144–1160, 2000.
- [23] D Q Mayne. Control of Constrained Dynamic Systems *. *European Journal of Control*, 7(2-3):87–99, 2001.
- [24] Tamás Keviczky, Francesco Borrelli, Kingsley Fregene, Datta Godbole, and Gary J. Balas. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on Control Systems Technology*, 16(1):19–33, 2008.

II

Thesis Appendices

A

Controller Parameters, Initial and Final Conditions

Table A.1: Scenario 1 - controller parameters.

		SL-AC	SL-RC and BL-RC	SL-DC	BL-DC
v_b	m/s	250	250	250	250
α	m/s ²	0	0	0	0
β	deg/s	± 0.5	± 0.5	± 0.5	± 0.5
N_p	-	16	4	4	4
c	s	60s	240s	60s	60s
W_f	-	0.0001	0.0001	0.0001	0.0001

Table A.2: Scenario 1 - initial and final conditions.

x_0	0	km
y_0	0	km
v_0	250	m/s
φ_0	90	deg
x_f	2500	km
y_f	0	km
δ_s	50	km
r_s	10	km
c_{sim}	1	s

Table A.3: Scenario 2 - controller parameters.

		SL-AC	SL-RC and BL-RC	SL-DC	BL-DC
v_b	m/s	180-200	200	180-200	180-220
α	m/s ²	± 0.1	0	± 0.1	± 0.1
β	deg/s	± 0.5	± 0.4	± 0.5	± 0.5
Np	-	24	4	4	4
c	s	50	300	50	50
W_α	-	0	0	0	10
W_β	-	0	0	0	10
W_f	-	0.0001	0.0001	0.0001	0

Table A.4: Scenario 2 - initial and final conditions.

x_0	0	km
y_0	0	km
v_0	200	m/s
φ_0	90	deg
x_f	1500	km
y_f	0	km
δ_s	50	km
r_s	10	km
c_{sim}	1	s

Table A.5: Scenario 3 with measurement updates and Scenario 3 with intermediate dynamic destination - controller parameters.

		SL-AC	SL-RC and BL-RC	SL-DC	BL-DC
v_b	m/s	180-200	180-200	180-200	180-200
α	m/s ²	± 0.1	± 0.1	± 0.1	± 0.1
β	deg/s	± 0.25	± 0.25	± 0.25	± 0.25
Np	-	30	6	6	10
c	s	60	300	60	60
W_α	-	0	0	0	10 - 0
W_β	-	0	0	0	10 - 0
W_f	-	0.0001	0.0001	0.0001	0 - 0.0001

Table A.6: Scenario 3 - initial and final conditions.

x_0	0	km
y_0	0	km
v_0	200	m/s
φ_0	90	deg
x_f	1500	km
y_f	0	km
δs	75	km
r_s	7.65	km
c_{sim}	5	s

Table A.7: scenario 3 with wider constraints, intermediate dynamic destination with and without measurement updates - controller parameters.

		SL-AC	SL-RC and BL-RC	SL-DC	BL-DC
v_b	m/s	180-200	180-200	180-200	180-220
α	m/s ²	± 0.1	± 0.1	± 0.1	± 0.1
β	deg/s	± 0.25	± 0.25	± 0.25	± 0.25
Np	-	30	6	6	10
c	s	60	300	60	60
W_α	-	0	0	0	10 - 0
W_β	-	0	0	0	10 - 0
W_f	-	0.0001	0.0001	0.0001	0 - 0.0001

Table A.8: Scenario 4 - controller parameters.

		BL-DC	BL-RC	SL-AC
v_b	m/s	180-200	180-200	180-200
α	m/s ²	± 0.25	± 0.25	± 0.25
β	deg/s	± 0.25	± 0.25	± 0.25
W_α	-	1	0	0
W_β	-	1	0	0
W_f	-	0.0001	0.0001	0.0001

Table A.9: Scenario 4 - initial and final conditions.

x_0	0	km
y_0	0	km
v_0	200	m/s
φ_0	90	deg
x_f	1500	km
y_f	0	km
δs	75	km
r_s	6.26	km
c_{sim}	5	s

Table A.10: Scenario 5 - controller parameters

		SL-AC	SL-RC	BL-DC
v_b	m/s	200	200	200
α	m/s ²	± 0	± 0	± 0
β	deg/s	± 0.5	± 0.5	± 0.5
Np	-	24	12	4
c	s	15	30	15
W_α	-	0	0	0
W_β	-	0	0	0
W_f	-	0.0001	0.0001	0.0001

Table A.11: Scenario 5 - initial and final conditions.

		AC-1	AC-2	
x_0	0	140	km	
y_0	0	0	km	
v_0	200	200	m/s	
φ_0	90	-90	deg	
x_f	300	-300	km	
y_f	0	0	km	
c_{sim}	5		s	

Table A.12: Scenario 6 - controller parameters.

		SL-AC	SL-RC and SL-BC	BL-DC
v_b	m/s	200	200	200
α	m/s ²	± 0	± 0	± 0
β	deg/s	± 0.5	± 0.5	± 0.5
Np	-	20	5	8
c	s	30	120	30
W_α	-	0	0	0
W_β	-	0	0	0
W_f	-	0.0001	0.0001	0.0001

Table A.13: Scenario 6 - initial and final conditions.

		AC-1	AC-2	AC-3	AC-4	
x_0	0	140	70.01	70.02	km	
y_0	0	0	-70	70	km	
v_0	200	200	200	200	m/s	
φ_0	90	-90	0	180	deg	
x_f	300	-300	70	70	km	
y_f	0	0	300	-300	km	
c_{sim}	5				s	

Table A.14: Scenario 7 and 8 - controller parameters.

		SL-AC	SL-RC	BL-DC
v_b	m/s	200	200	200
α	m/s ²	± 0	± 0	± 0
β	deg/s	± 0.5	± 0.5	± 0.5
Np	-	24	8	6
c	s	10	30	10
W_α	-	0	0	0
W_β	-	0	0	0
W_f	-	0.0001	0.0001	0.0001

Table A.15: Scenario 7 and 8 - initial and final conditions.

	AC-1	AC-2	
x_0	0	140	km
y_0	0	0	km
v_0	200	200	m/s
φ_0	90	-90	deg
x_f	140	0	km
y_f	0	0	km
c_{sim}	5		s

B

Scenario 2, Level of detail comparison

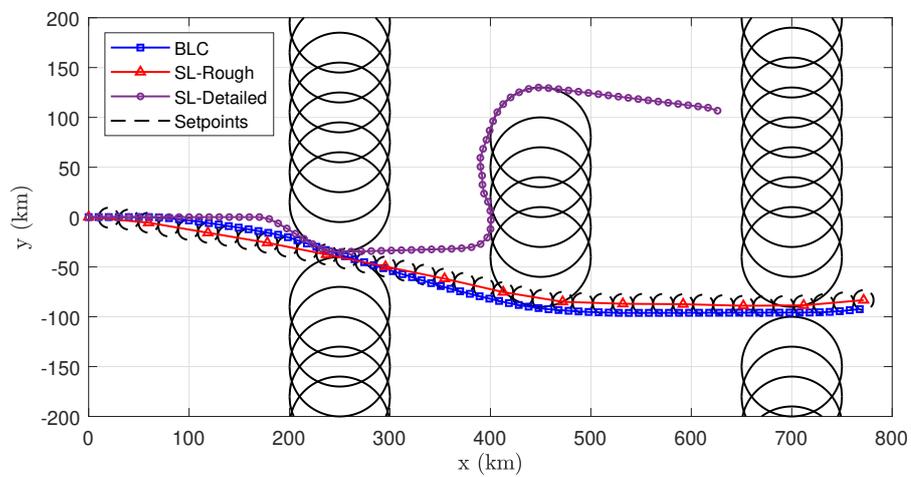


Figure B.1: Trajectory comparison of the BLC with its contributing SL-RC and SL-DC for scenario 2.

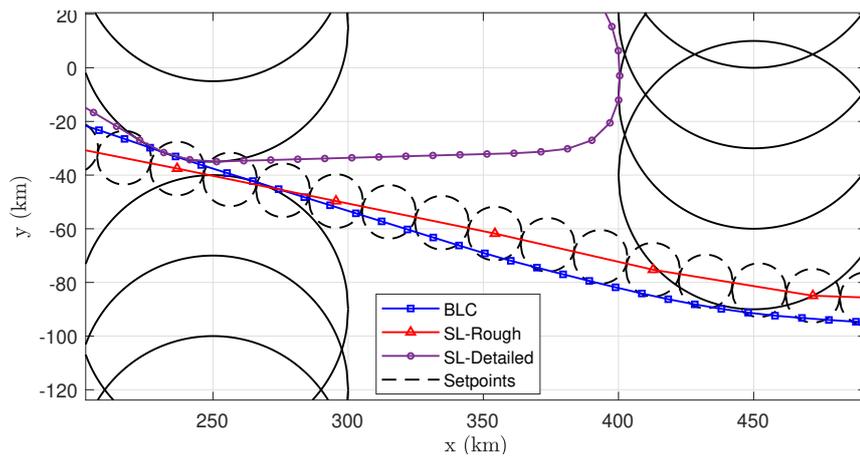


Figure B.2: Close up of Figure B.1, showing the trajectory around a forbidden area.

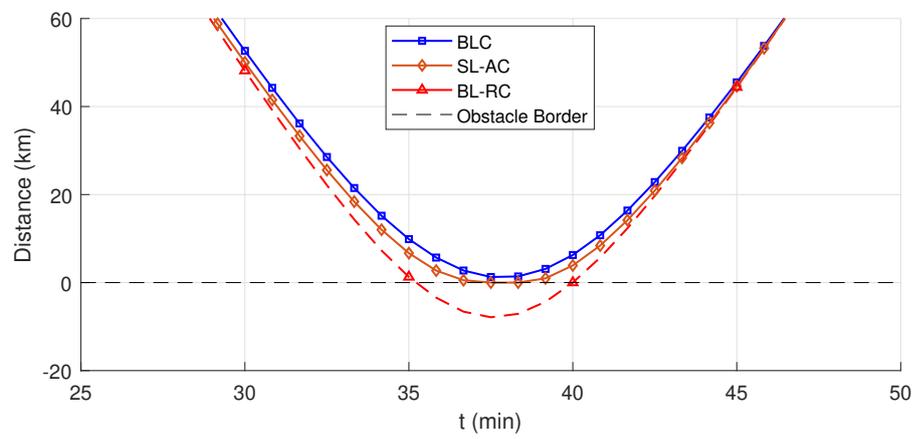


Figure B.3: Positional relative distance to the forbidden area boundary for scenario 2.

C

Scenario 4 Trajectories

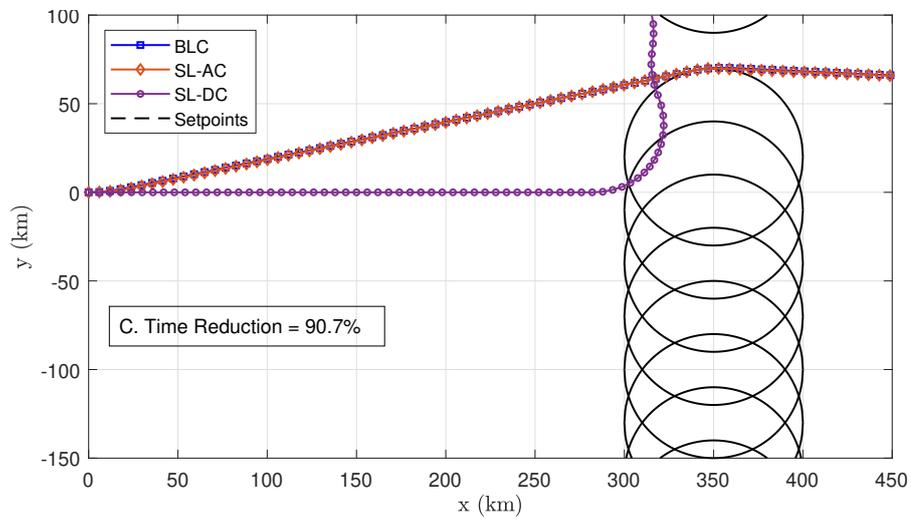


Figure C.1: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 60$ s, scenario 4.

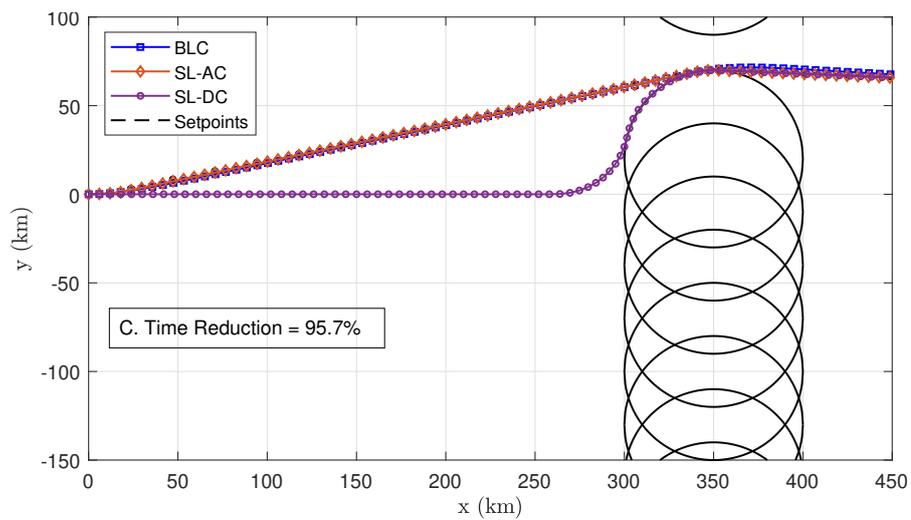


Figure C.2: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 120$ s, scenario 4.

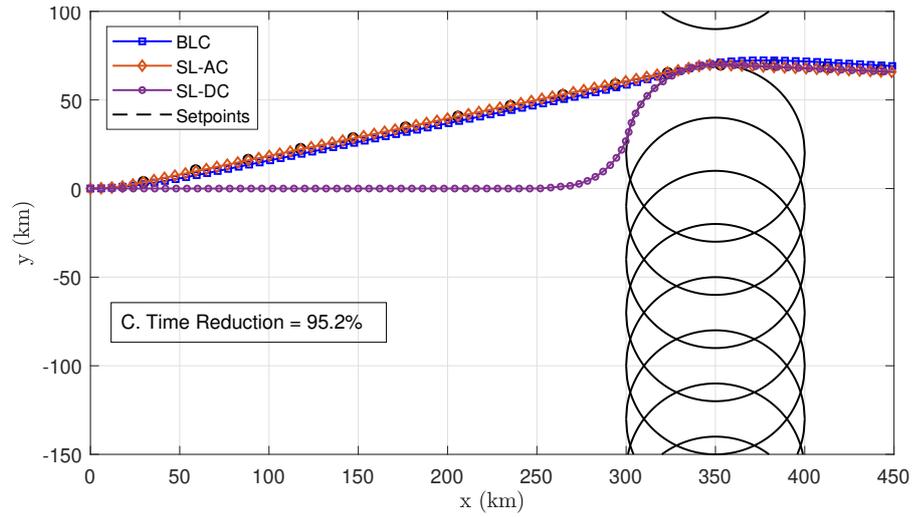


Figure C.3: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 150$ s, scenario 4.

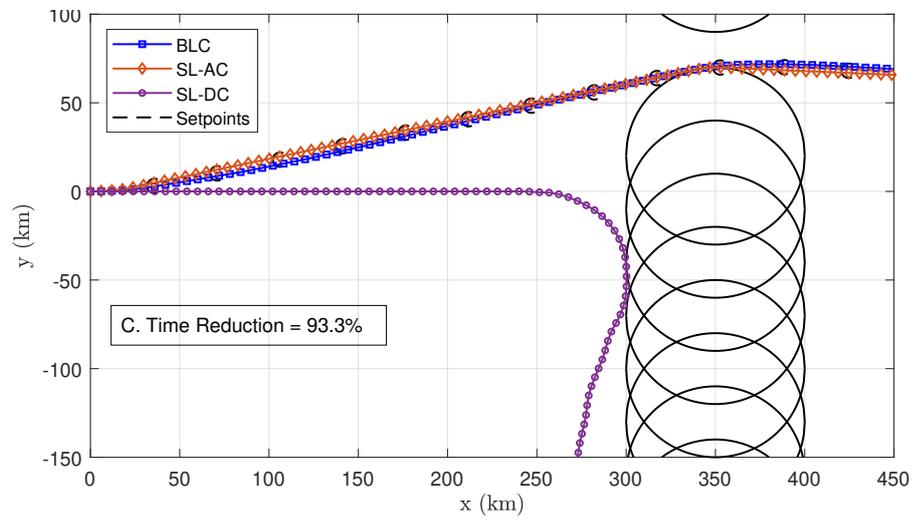


Figure C.4: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 180$ s, scenario 4.

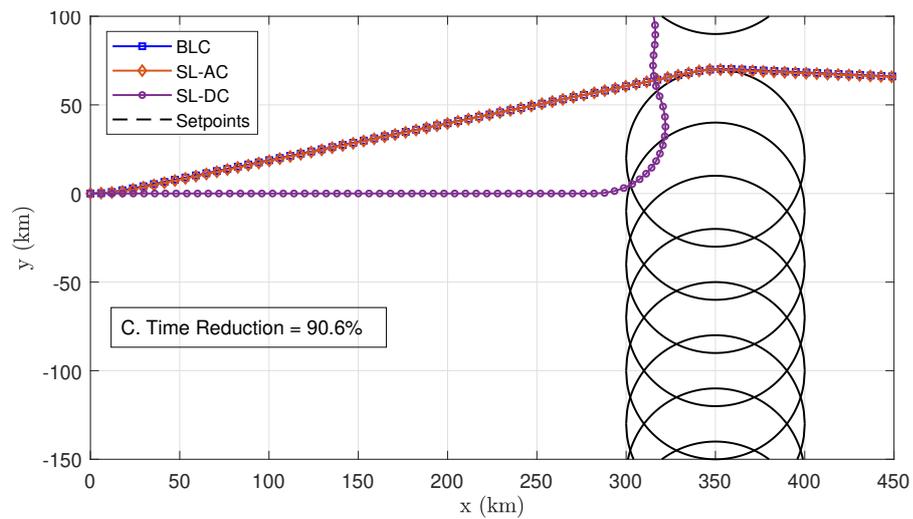


Figure C.5: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 60$ s and $c_s = 30$ s, scenario 4.

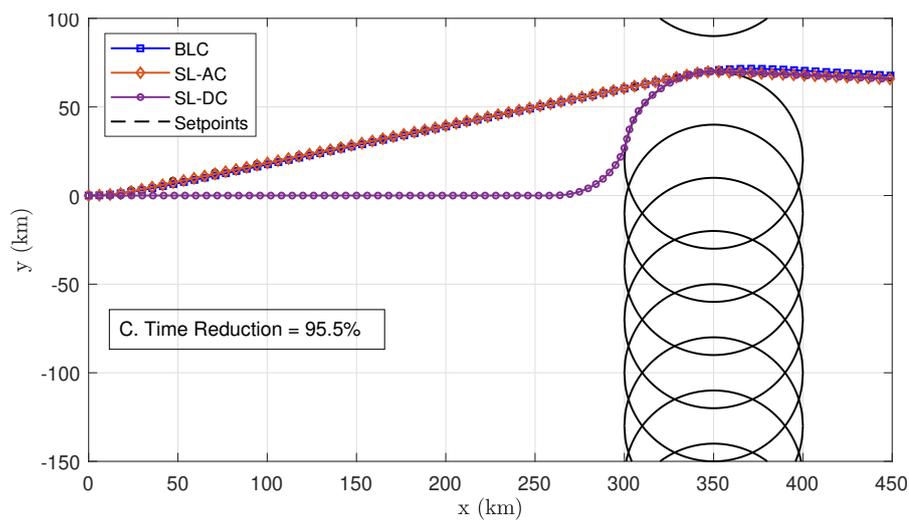


Figure C.6: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 120$ s and $c_s = 60$ s, scenario 4.

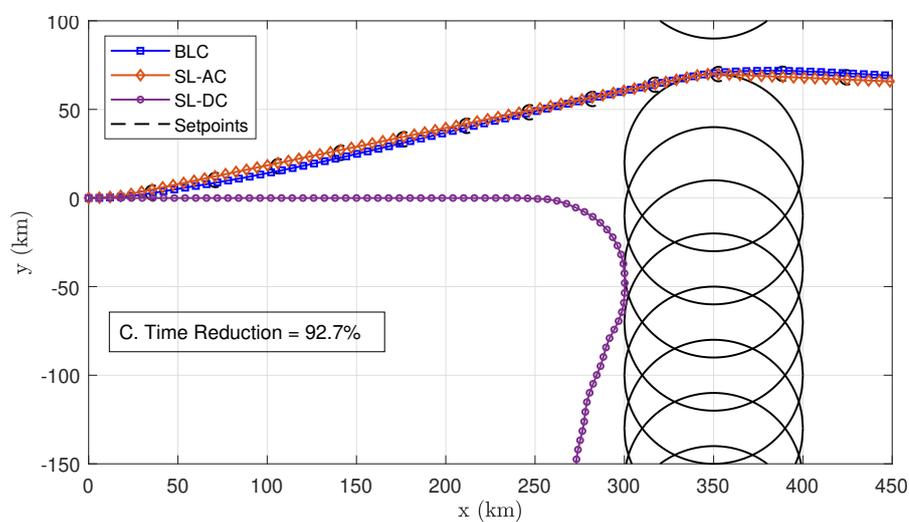


Figure C.7: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 180$ s and $c_s = 90$ s, scenario 4.

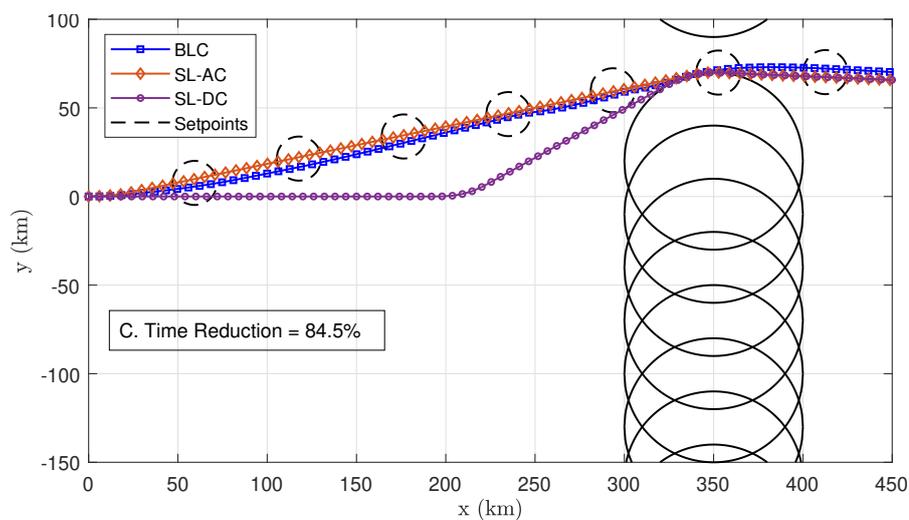


Figure C.8: BLC, SL-AC and SL-DC trajectories comparison with $c_r = 300$ s and $c_s = 150$ s, scenario 4.

III

Preliminary Report & Literature Study
Already Graded

1

Introduction

EUROCONTROL predicted in 2010 that there are likely to be between 13.1 and 20.9 million Instrument Flight Rules (IFR) flight movements in 2030, which is 1.4% to 2.2% more than in 2009 [1]. The forecast by EUROCONTROL in 2018, is that a total growth of 53% in 2040 is expected compared to 2017 [2]. An ever-increasing rise in the workload for air traffic controllers as well as further saturation of certain air spaces is imminent if innovations in air traffic control are not implemented. Besides a reduction of workload for air traffic controllers, reducing flight time and minimizing aircraft emissions are other reasons to strive for more efficient air traffic control.

This research project aims to design and test a model predictive controller (MPC), with a bi-level hierarchical architecture to optimize global air traffic in terms of the flight time and total fuel consumption. The bi-level control architecture will be developed and evaluated using Python and BlueSky. The main contribution of this thesis is to investigate the integration of a centralized model predictive control level with a decentralized model predictive control level, to obtain a single coherent controller by the formulation of the integrating link and the communication flow. The added benefits in terms of safety, fuel consumption and flight time will be explored.

1.1. Project Scope

The research scope includes the development of two model predictive controllers and efficient integration of the controllers. The second part of the research is focused on evaluating the performance of the developed bi-level control system in terms of feasibility, control performance and computation time. Moreover, the task allocation and cooperation between the two control levels for achieving the global goals are designed and tested. The novelty of the project lies in the combination of decentralized controllers with limited global knowledge, a smaller prediction horizon, with a fast control sample time, and a centralized controller with global knowledge of the system and a larger prediction horizon, but a slower control sample time.

The bi-level controller will be applied to flight movements in the horizontal two-dimensional plane. The system will be tested using Python and BlueSky simulations. Optimizing the computational efficiency of the bi-level controller to achieve real-time control is desirable but not a focus point of this research project. Furthermore, the parallel computation of the centralized and decentralized control levels will not be part of this thesis. Proving stability, robustness and the assessment of global optima fall outside the research scope of this project. Furthermore, the separation of aircraft will be conducted by the centralized controller. The decentralized aircraft separation is a topic for future research.

1.2. Research Questions

The research question and sub-questions presented in this section must be answered to reach the goal of the research project. The main research question of the thesis is:

“How can a synergy be found between a fast-rate and slow-rate model predictive controller for air traffic control applications, where the benefits of each controller are combined to improve the safety, fuel consumption and flight time compared to a single level model predictive controller?”

Ultimately, the two controller levels will be integrated and the following questions must be answered.

What information will be shared between the control levels? At what rate will the information be shared? And most importantly, how will the output of the centralized controller be formulated for the decentralized controllers? To answer all aspects of the question, the sub-question can be formulated more specifically as follows.

The limitation of the system on the physical system must be converted to optimization constraints. Giving rise to the question, what are the system constraints and what effect do they have on the controller in terms of computation time and feasibility? Do conflicts arise between the model constraints and the minimum separation constraints in high-density traffic scenarios?

Additionally, a balance needs to be found between the sample time and the prediction horizon for the centralized and decentralized controllers. These design choices might affect the optimization solution as well as the computation time. Furthermore, the cost function weights are a design parameter that must be tuned to balance fuel consumption with the flight time. Therefore, research must be done, answering the question, how are the computation time and the performance affected by tuning the prediction horizon and control sample time?

Finally, to answer the performance-related part the following questions need to be answered, what are acceptable computation times for the centralized and decentralized controllers for safe operation of each controller? What effect do the cost function weights have on the solution? What is a proper trade-off between computation time and performance requirements and how can a trade-off between flight time and fuel consumption be made?

1.3. Research Objectives

The research objectives of this project consist of two parts. The first part of the project entails the development phase of a bi-level model predictive controller for air traffic management. The second part of the project focuses on testing the performance of the controller by conducting case studies. The case study experiments will provide quantitative answers to the research question and aim to motivate and support the use of multiple model predictive control levels. The first main research objective of this thesis is:

“Optimize the efficient use of airspace, fuel consumption and flight time by the design, development, and integration of a decentralized model predictive controller with limited global knowledge, a smaller prediction horizon, a fast control sample time and a centralized model predictive controller with global knowledge of the system and a larger prediction horizon, and a slower control sample time”

Once both levels of the controller are operational, the controllers will be integrated by achieving a method cooperation between the two control levels to benefit from each controller's strengths and compensate for each others' weaknesses by developing a strategy of collaboration. The second essential research objective of this thesis is:

“Determining the benefits of a bi-level model predictive controller performance in terms of cost function optimization, computational efficiency, and feasibility by comparing it to a single centralized or decentralized model predictive controller using case studies in Python and BlueSky.”

The two main research objectives give rise to several sub-objectives that contribute to fulfilling the main objective.

“Developing a centralized controller, decentralized controller, and the integrating link by formulating the optimization problem and using Python to develop the system.”

The second objective is focused on testing and tuning the bi-level controller which is divided into three sub-objectives. The developed control system will be tested in a series of case studies, where an aircraft will start at a certain position and will fly towards a final destination.

“Testing the performance of the bi-level controller in comparison to a single level model predictive controller on a single aircraft by using case studies in which the aircraft flies a trajectory towards a given destination while avoiding static obstacles.”

The performance of the controllers will be evaluated and compared based on the total flight time, distance traveled and fuel consumption. The prediction horizon and the control sample times are used as independent variables. The third sub-objective is concerned with aircraft separation:

“ Testing the capability of the controller to safely manage conflicting aircraft by creating conflict scenario's where two identical aircraft have to cross each other to reach their destination with varying relative heading angles.”

This sub-objective provides an understanding of how the bi-level controller deals with conflicts and manages aircraft separation during flight. Conflict free trajectories are of vital importance in air traffic control applications and will be addressed extensively. Finally, the bi-level controller will be tested in a multi-agent system.

“Evaluating the performance and feasibility of the bi-level model predictive controller in terms of the number of conflicts by conducting case studies using multiple heterogeneous aircraft with varying initial conditions and destinations.”

To test the performance in a more realistic scenario, the controller will finally be applied to several case studies with an increased traffic density. The number of aircraft that will be simulated and controlled simultaneously depends on the computational efficiency of the controller since it is expected that an increase of aircraft to the system can exponentially increase the computation effort required.

1.4. Preliminary Report Structure

In this preliminary thesis report first, a literature survey is provided. Chapter 2, focuses on the working principles of model predictive control. In Chapter 3, an overview of decentralization and several conflict detection and avoidance research projects related to air traffic management is provided. Followed by, a literature review concerning decentralized and distributed control as well as a review on multi-level control architectures, in Chapter 4. The conceptual description and conceptual design of the controller can be found in Chapter 5. In Chapter 6, the project plan, as well as, the development plan of the bi-level model predictive controller, is presented. The experimentation and testing plan is provided in Chapter 7. Finally, in Chapter 8, the preliminary results obtained during the first experiment phase are presented.

2

Model Predictive Control Working Principles

Model predictive control, also referred to as receding horizon control (RHC), has increasingly been used in the past decades in many different fields of engineering, from automotive [3][4] to industrial processes [5][6], maritime [7] and also aviation [8][9][10][11]. In a survey paper from 1989 [12], an extensive history of early applications of MPC is described.

The working principles of MPC are explained in Section 2.1. The characteristics that define the air traffic optimization problem are described in Section 2.2. Finally, the aircraft model used during this project is presented in Section 2.3.

2.1. Working Principles of Model Predictive Control

In this section, the general working principles of MPC are explained. In the recent paper by Orukpe [13], a comprehensive approach to the basics of MPC is described. Much of this chapter is from the paper by Orukpe and two textbooks on MPC [14][15].

Model predictive control is a control strategy capable of achieving an optimal control input, minimizing a certain cost function. MPC uses a model of the system dynamics to predict the future states of a system, for a certain control horizon H_c , within a given prediction horizon H_p . The control horizon can be equal to, or smaller than the prediction horizon. When the optimization is complete, the first control input of the optimal input sequence is applied to the controlled system, and the remainder of the input sequence is disregarded. The time window shifts one time step further and the optimization problem is repeated, based on new state measurements. The controller input sequence is given by Equation 2.1.

$$\mathbf{u}(k) = [u(k|k), u(k+1|k), \dots, u(k+H_c-1)] \quad (2.1)$$

The working principle of MPC is schematically represented in Figure 2.1. The input sequence is given by the purple dashed line. The orange line represents a new control input sequence, determined one time step further in time, based on newly available measurements.

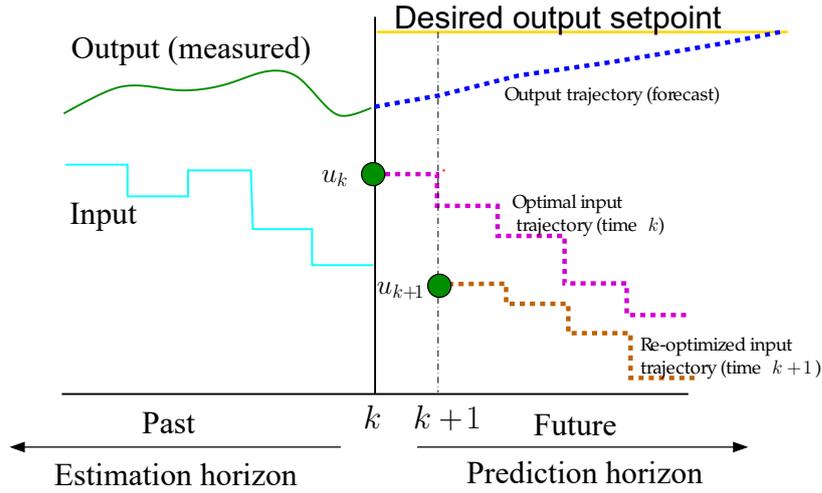


Figure 2.1: Graphical representation of the moving horizon principle [16].

2.1.1. Prediction Model

Since a model predictive controller makes a forecast about the future states to determine the current control inputs, a model is required to predict the state trajectories. It has to be noted that, small discrepancies in the model can generate large accumulating errors over time. Therefore, future control inputs at a later stage of the prediction horizon become increasingly unreliable. Consequently, increasing the prediction horizon time does not necessarily yield better control performance, however, it does raise the computational effort due to the additional decision variables.

In Section 2.3, the kinematic aircraft model that is used is explained. The same model is used on both levels of the bi-level control scheme, in a generalized case, these models can vary. Because a deterministic model is used during the simulations, a wind field can be enabled to generate noise in the system.

2.1.2. Cost Function

The cost function typically consists of multiple terms that are minimized for a prediction horizon. These parameters can be the error between a reference state and the measured states, the intensity of the control input, the variation between consecutive control inputs, overshoot, rise time or the minimization of resource consumption like fuel or flight time.

Each factor in the cost function can be assigned a weight, to put more emphasis on specific parameters, and tune the controller. A practical example regarding air traffic would be that a pilot can choose a certain balance between fuel consumption and flight time. Tuning the cost function parameters proves to be one of the main challenges of designing a model predictive controller [15]. The cost function for the controller proposed in this research can be found in Chapter 5.

2.1.3. Constraints

A major advantage and key feature of MPC is the possibility to imply constraints on the input, output, and states of the system that is being controlled. This is different from other optimal control methods such as linear quadratic regulator (LQR) control. Input constraints can be the physical limitation of the gas pedal of a car, or the actuators of an aircraft, for example. Whereas the output constraint usually represents desired boundaries. Constraints on the inputs are often hard constraints as they are based on the physical limitations of a system.

Constraints on the output or states can be designed to be soft, a soft constraint can physically be violated, due to disturbances, for example. A hard constraint is typically a physical system constraint such as maximum acceleration, or stall speed. When designing for global air traffic, constraints also apply to safety regulations, such as minimum required separation between any two aircraft. The input constraint boundaries are described by,

$$u_{\min} \leq \mathbf{u}(k) \leq u_{\max} \quad \forall t \quad (2.2)$$

the output constraint on any state is formulated as,

$$y_{\min} \leq \mathbf{y}(k) \leq y_{\max} \quad \forall t \quad (2.3)$$

and the separation between two aircraft is constrained using,

$$\sqrt{(\Delta x_{ij}(k))^2 + (\Delta y_{ij}(k))^2} \geq \delta \quad (2.4)$$

The implementation of separation constraints has been a research topic for engineering fields outside aviation as well, like in maritime engineering [7]. In the paper by Ferranti et al. [7], MPC is used to optimize multi-robot trajectories while maintaining safe separation between agents. A centralized method for finding the global optimum by a single coordinator is said to raise difficulties. All agents in the system are required to share, possibly classified, information (dynamics, performance models, product constraints, etc), with the centralized node. They propose to solve this problem by not relying on a centralized coordinator, instead, the problem is distributed over local agents.

Ferranti et al. state that a solution can be that every agent transmits and receives the predicted position to and from neighboring agents. These predicted positions can be used as constraints to solve the local optimization problem to find a conflict-free trajectory. However, the authors avoid this method since the predicted and actual location between two-time instances can vary significantly. They propose the alternating direction of the method of multipliers (ADMM) to handle the coupled constraint in a distributed manner.

2.1.4. Advantage and Disadvantages of Model Predictive Control

The popularity of MPC for optimal control applications is largely due to its ability to handle both physical and control constraints. Furthermore, it is an optimal control strategy and MPC can be applied to control linear and non-linear systems with linear or non-linear models, constraints, or cost functions.

Nevertheless, stability and robustness theorems are not as well developed for non-linear systems as they are for linear systems [17]. Stability theories for linear MPC are relatively mature, yet the non-linear variant still requires more research [18].

It is difficult to theoretically prove the stability and performance of non-linear models when used in the context of MPC [19]. The optimal control problem requires a finite horizon to be able to be solved repeatedly online. However, to ensure stability, an infinite horizon is necessary [20].

A disadvantage of MPC is that the computation time can quickly become a limiting factor for fast real-time applications, especially when applied to large scale problems such as air traffic control. MPC requires a high computational capacity because the optimization algorithm is iteratively repeated, which made it only feasible for slow processes in the early days.

2.2. Characteristics of Air Traffic Control Optimization

It is essential to identify what kind of optimization problem is faced, and how an air traffic control problem can be characterized in terms of optimization and what the characteristics of the model are. Linear or non-linear, smooth or non-smooth, convex or non-convex. If the MPC is based on convex quadratic optimizations using linear constraints, the computation time can be improved tremendously [21].

The optimization problem is combinatorial, the size of the airspace and the number of aircraft give the problem a large optimization space [22]. The two main challenges of optimizing flight paths are that the problem is non-convex and non-linear, yielding a heavy computational burden [23]. Analytic solutions are generally not available, except for specific hypothetical cases. Therefore, the numerical burden is often the main driving factor for practical implications [21].

Compromises have to be made on the optimality of the solution, a non-perfect, conflict-free solution might have to be accepted as sufficient [24]. Using a bi-level structure, the compromise between optimality and computation time can be avoided and both, the optimality as well as an acceptable computation time can be assured by reducing computation time and sacrificing optimality at the centralized control level and regaining optimality at the decentralized control level. The computation time of the centralized optimization can be reduced by applying a larger control sample time resulting in a loss of detail. The decrease of detail can be compensated by the introduction of a decentralized model predictive control level, with a faster control sample time.

Considering that the model, the constraints and the cost function are non-convex, the solution that is found by the controller can be a local minimum instead of the global optimal solution. Proving that an optimal solution is the global optimum is difficult. For this particular air traffic control application, finding a local optimum, is acceptable, given that all safety and physical constraints are satisfied. A multi-start solution whereby the optimization is reiterated with different initial conditions several times could result in finding the global optimum. A multi-start solution requires that multiple optimization cycles are computationally possible within the control sample time, or it requires parallel computing. Applying a multi-start solution to the controller is a topic for future research.

2.3. Aircraft Model

The aircraft model presented in this section is based on the equations used in the BlueSky 'Open Aircraft Simulator'. Since the same model equations are used by the MPC as the simulation software, parameter discrepancies and wind is introduced to make the simulation more realistic. The model is an integrated model, consisting of two parts. The first part is used to determine the position and velocity of the aircraft. The second part is used to compute the fuel flow. This model is based on the OpenAP Aircraft Model developed by Sun et al. [25]. The position, heading, and velocity of each aircraft is determined by using Equations (2.5-2.8) and are used for both levels of the controller.

$$V(k+1) = V(k) + a(k)\Delta k \quad (2.5)$$

$$x(k+1) = x(k) + V(k) \sin(\varphi(k))\Delta k + W_x\Delta k \quad (2.6)$$

$$y(k+1) = y(k) + V(k) \cos(\varphi(k))\Delta k + W_y\Delta k \quad (2.7)$$

$$\varphi(k+1) = \varphi(k) + g_0 \frac{\tan(\psi(k))}{V(k)} \Delta k \quad (2.8)$$

The equations determining the fuel consumption (2.9-2.13) used in this model are excerpted from BlueSky.

$$C_L = \frac{m \cdot g_0}{\frac{1}{2} \rho V(k)^2 S} \quad (2.9)$$

$$D = \frac{1}{2} \rho V(k)^2 S \cdot (C_{D,0} + k_0 \cdot C_L^2) \quad (2.10)$$

$$T = D + m \cdot a(k) \quad (2.11)$$

$$T_{\text{ratio}} = \frac{T}{N_{\text{eng}} \cdot T_{\text{static}}} \quad (2.12)$$

$$\sum F_k^{k+1} = N_{\text{eng}} \cdot (A_f \cdot T_{\text{Ratio}}^2 + B_f \cdot T_{\text{ratio}} + C_f) \Delta k \quad (2.13)$$

The control inputs acceleration and bank angle are piece-wise constant, and applied as a zero-order-hold (ZOH) function, as shown in Figure 2.2.

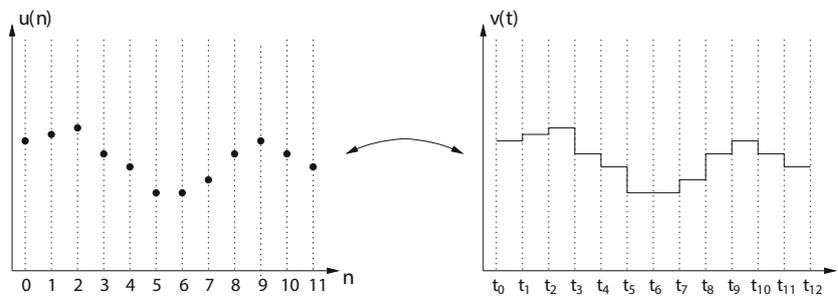


Figure 2.2: Zero-order hold approach to piece-wise continuous control inputs [26].

3

Decentralization in Air Traffic Management: Motivation and Concepts

The main reasons for an interest in next-generation air traffic control methods are the rapid growth in total flights and the projected saturation of certain airspaces in the coming decades [1][2]. It was already predicted in 2006, that air traffic would double by the year 2025 [27]. In [28], it is mentioned that a vision for the European airspace was to have approximately 16 million flights per year, by the year 2020, while also reducing fuel use and flight time. One of the limiting factors to achieving this goal is the workload of air traffic controllers. Two institutions working on next-generation air traffic control are; NextGen in the USA and it's European Counterpart SESAR, an overview of both organizations can be found in [29].

Many varieties of techniques have been proposed as potential next-generation air traffic control solutions. The survey paper from 2000, by Kuchar et al. provides an overview of Conflict Detection & Resolution (CD&R) and Conflict Detection & Avoidance (CD&A) from before the year 2000 [30]. Genetic algorithm techniques have been developed in [23][31][32], these algorithms generally struggle with long computation times and scalability issues. Potential field algorithms considering aircraft as charged particles that repel each other can be found in [33][34], an effective method for free flight, however determining the level of safety is difficult due to decentralization. Research in mixed-integer (non-)linear programming (MILP) is also being conducted for airspace control applications [35][36][37]. For multi-vehicle coordination, the global optimal solution scales poorly because the optimization quickly becomes complex for large fleets when using quadratic programming and MILP. Game Theory has been applied for air traffic distributed control in [38], while this research has found a promising result, the efficiency did decrease as more aircraft were added to the system.

Constraint programming approaches such as model predictive control are researched in [24][8][9][10][11], and will be elaborated in more detail in this chapter. In the past decade, research in next-generation air traffic control has also extended to drones and small unmanned vehicles and how these aircraft can safely operate without risking interference with air traffic [21][39][40]. Analytic solutions are generally not available, except for specific hypothetical cases. Therefore, the numerical burden is often the main driving factor for practical implications [21].

The introduction section of [22] provides a good summary of recently proposed applications. A selection of these research fields involving 4D trajectories, optimization-based approaches, and decentralized or distributed multi-agent system approaches will be explained in this chapter. In Section 3.1, the difference between, centralized, decentralized and distributed air traffic control methods is discussed. Several Conflict Detection & Avoidance concepts will be mentioned in Section 3.2, to give an overview of the lessons learned in this research field. Methods utilizing 4D trajectory prediction are presented in Section 3.3. Finally, in Section 3.4, optimization-based approaches will be highlighted.

3.1. Centralized, Decentralized and Distributed Air Traffic Management

Two main philosophies can be distinguished in this research field; centralized and decentralized control solutions. This section aims to discuss the differences between the two groups and to reflect on the advantages, disadvantages and the current status of developments.

Today's tactical air traffic control system relies on the abilities and skills of the air traffic controllers. There is little automation in ensuring adequate separation between aircraft. The system which is currently in place can be considered safe, but the use of airspace is not as efficient as it could potentially be [41].

Centralized control approaches operate with a centralized node that collects and processes all the information in the network. These methods aim to separate air traffic for the entire duration of the flight by making strategic flight plans. The current ATC situation, where an ATC center manages multiple aircraft above a particular sector can be considered a centralized approach. Since a centralized agent has knowledge of the entire system, it can predict and find the best global solution and is mathematically speaking the best method.

Decentralized solutions are the alternative option, where each aircraft can determine its preferred route while maintaining an adequate safe distance between its direct neighbors. In the past this was often referred to as 'Free-Flight', currently, decentralized or distributed air traffic control is more commonly referred to as conflict detection and avoidance.

Figure 3.1, shows the difference between a decentralized air traffic structure where each aircraft is controlled locally and a centralized air traffic structure where a single node controls the full region.

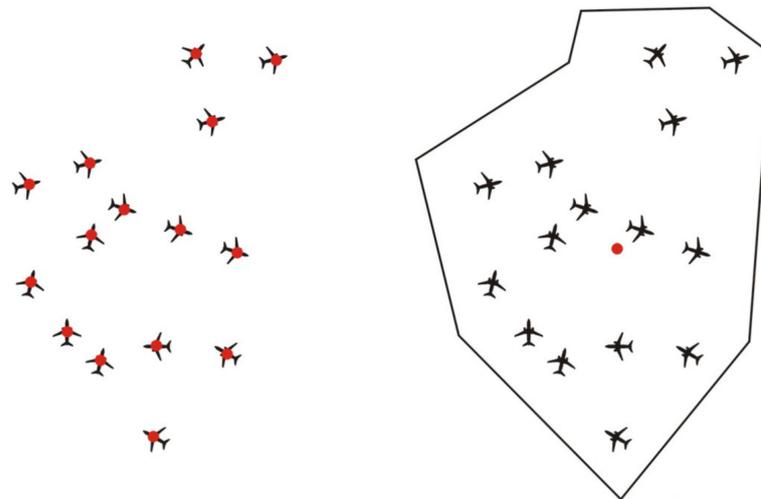


Figure 3.1: Decentralized air traffic control structure (left) compared to a centralized air traffic control structure (right) [42].

The Automatic Dependent Surveillance-Broadcast (ADS-B) system is currently installed in nearly every aircraft and is a system that potentially makes distributed air traffic feasible. The system enables airplanes and ground stations to broadcast and receive information from other aircraft. The ADS-B unit automatically transmits, the equipped aircraft's, position, predicted route and velocity along with the aircraft identifier. All neighboring aircraft can receive this information and are therefore aware of the presence and intents of other aircraft.

From a mathematical perspective, decentralized concepts are usually less optimal compared to a perfect centralized optimized solution in terms of cost function optimization. Yet, the computation times are generally not an issue in decentralized flight solutions. Due to the distribution of the global problem over multiple agents, the computational effort is also divided over several agents. Centralized optimizations for systems as complex and as broad as global air traffic control, become computationally extremely expensive and next-to-impossible with current hardware to use in real-time.

From the review by Al-Gherwi et al. [19], it becomes clear that only a centralized model predictive control unit with one overall objective function for the entire system results in guaranteed optimal performance. When the system is fully decentralized with independent local models and with independent local objectives functions for each subsystem, there is an expected loss in optimal performance.

The uncertain nature of global air traffic prediction is a downside of global optimization that local free-flight solutions do not have a significant problem with, due to the shorter time horizons. Furthermore, the low predictability over the full length of a flight path is another issue rendering the global optimization solution

far from ideal. Currently, predictions are still too inaccurate and short term traffic resolutions require human expertise [24]. The low predictability of the problem demands near real-time updates of the optimization problem, computationally practically impossible with the current technology.

Besides the computational complexity of centralized optimization problems, another issue is that a system with a single centralized controller is prone to catastrophic failure due to it being a single point of failure. A risk that is associated with a fully centralized control architecture is that a failure at the centralized level results in a failure of the whole system whereas failure of one of the decentralized controllers can be covered by the still operational controllers mitigating the risk of single points of failures and making the overall system more redundant. Therefore, for a fully centralized system, another centralized, free-flight or decentralized control system is necessary as a back-up [42].

3.2. Conflict Detection & Avoidance Concepts

Collision avoidance is the most important aspect of air traffic control due to safety being the top priority for air traffic. Any concept that aims to improve the current system in terms of optimality or efficiency must be at least as safe as the accepted safety in the current system, or preferably even safer. For aviation, the minimum lateral safety constraint is at least 5 nautical miles. The vertical safety regulations state that the vertical spacing should be a distance of at least 1000 ft, see Figure 3.2.

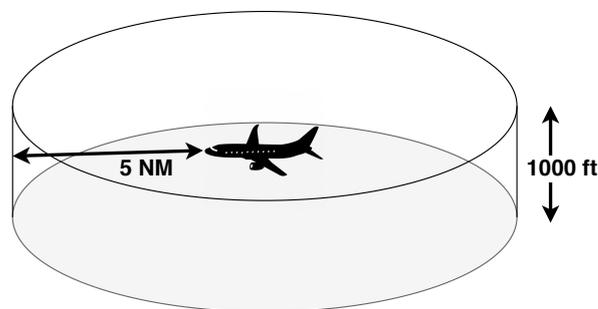


Figure 3.2: Minimum safety margins around en-route aircraft.

In this section, several recent CD&A concepts that are closely related to the proposed concept for this thesis are explored. Such a concept is the ERASMUS project. ERASMUS (En-Route Air Traffic Soft Management Ultimate System) is a system developed in 2004 that organizes air traffic in such a way that future conflicts are avoided by slightly changing aircraft speed on a time horizon of approximately 15 minutes. ERASMUS aims to reduce the workload of air traffic controllers and therefore increasing the capacity of the airspace. The algorithm works alongside air traffic controllers and makes it easier for them to manage the total air traffic [32]. The ERASMUS project provided global optimization methods to achieve subliminal speed control. The ERASMUS project uses a genetic algorithm, which can still be extremely time consuming when not making concessions in the required accuracy or shortening the time horizon.

Another interesting approach is proposed in [43], where the authors propose a multi-agent system in which altitude is constant during the cruise phase. Airlines are free to choose a Cost Index (CI), to adjust the priority from reducing fuel cost to reducing flight time. The CI setting is an important parameter that can be used to adjust the weights of the cost function in an optimization algorithm. For the simulations in this paper, the velocity is constrained to stay within a bound of -6% and +3% of the optimum cruise speed at a particular altitude of a certain aircraft type. For this research, the Airbus A320 and A380 were used, which have an optimal speed of 446 knots and 487 knots at 36000 ft, respectively. Aircraft acceleration and deceleration are fixed at 0.572 m/s^2 and a standard turn radius is set at $3^\circ/\text{s}$. These values are used as a guideline to determine approximate input constraints for the bi-level controller in this thesis research.

The multi-agent algorithm proposed in [43], only minimizes the number and duration of conflicts but is not capable of solving everything. An air traffic controller is still required to provide additional resolution maneuvers to resolve all the conflicts. A greedy algorithm is used to perform a local optimization process, where at each time step an aircraft can only accelerate, decelerate or cruise. It may occur that local optimization cannot find a conflict-free solution. The problem can be reduced by adding intermediate decisions, at the cost of adding additional computation time. However, the goal of their research is to show the benefit of

using a multi-agent system in air traffic control. The multi-agent system is capable of solving about 86% of the conflicts by only using subliminal speed control and is capable of dealing with non-cooperative agents in the system. In reality, altitude and heading changes can be used as extra degrees of freedom to de-conflict situations. This is a promising result for multi-agent systems but needs more research in terms of real-time applications and full conflict-free solutions.

3.3. 4D-Trajectory Methods

In 4D trajectory oriented systems, conflict-free trajectories are generated for each aircraft. These trajectories can be used for scheduling and sequencing as well as for early conflict detection and resolution, rerouting and arrival time management [44]. A general complication for trajectory-based concepts is that trajectories are determined at an early stage and have to make predictions far into the future. Due to inaccuracies in aircraft performance models, discrepancies in model parameters such as the aircraft weight, drag polar and engine performance data, uncertainties will inevitably occur over the vast time horizons. Furthermore, unpredictability's in changing weather conditions, pilot errors, measurement errors, and precision problems in the aircraft equipment all contribute to the complexity of full trajectory optimization [41]. To account for uncertainties, additional buffers need to be placed on the required separation between the predicted positions of aircraft. This leads to inefficient use of airspace since these buffers required more airspace than necessary, inefficient flight paths can be the result [44].

4D trajectory oriented system can be considered a centralized system because a single agent provides a full trajectory for all aircraft. Some 4D trajectory concepts are: the Programme for Harmonized Air traffic Management in Europe (PHARE) [45] and Distributed Air/Ground Traffic Management (DAG-TM) [46]. A brief description of PHARE and DAG-TM can be found in the introduction section of a paper aiming to combine 4D trajectory optimization with local airborne separation assistance [44]. The PHARE program ended in 2000 and failed due to the lack of the technology necessary to make the contractual '4D Tubes' reality and the lack of conflict resolution implementation. The project did conclude that a trajectory-based approach could potentially work [42]. Throughout the proposed Free-Flight and 4D-trajectory flight concepts in literature, it is often mentioned that a combination of multiple control levels is a likely future scenario to resolve these issues [42].

A combination of strategical 4D-trajectory planning with tactical air traffic systems, to increase the efficient use of airspace while maintaining safety is proposed in [41][47][44][45]. The authors of [41], propose a system of multiple levels where tactical aircraft resolution is done by air traffic controllers and strategic trajectory planning is performed using trajectory-based concepts, combining the best of both worlds. A concept that combines Airborne Separation Assistance Systems (ASAS) with trajectory-based operations to mitigate the uncertainties of the flight trajectories by making sure the controlled aircraft stay separated by a different system is described in [44]. The benefits mentioned in this paper are a reduction in controller workload, limited deviations from the 4D path and minimization of the long term prediction uncertainty. Minimize route or altitude changes for local separation assurance, While minimizing the impact on the flight crew. Furthermore, the benefits of reducing buffers while still using full trajectory optimization are acknowledged. Resulting in an ATC architecture in which aircraft are free to optimize their own globally conflict-free path by taking advantage of traffic flow management. Inaccuracies and possible conflicts are detected and resolved by ASAS.

3.4. Optimization Based Decentralized Control in Air Traffic Management

In recent research regarding next-generation air traffic control applications, model predictive control has been used more often. The ability to optimize non-linear systems as well as the ability to seamlessly add constraints to the optimization problem it a suitable technique for air traffic control, which is a constrained, highly non-linear problem. A method to mitigate a full-scale optimization problem is to use distributed or decentralized model predictive control. Decentralized control makes use of the fact that there are many agents or in this case, aircraft in the system, that can all contribute to finding a global (sub-)optimal feasible solution in real-time.

In this section, multiple air traffic management concepts applying model predictive control or receding horizon control for trajectory optimization or conflict resolution are highlighted. One research project where

(distributed) MPC is used is described in [48][49]. A system with multiple agents with decoupled dynamics can be coupled with the cost function or through the constraints. Within the air traffic control field, the dynamics of the aircraft are decoupled, but the constraints are usually shared among the different agents [49]. When no centralized agent is present, like a fully decentralized or distributed system, an optimization sequence needs to be defined. A predefined sequence is proposed in some research papers, however, these solutions are not scalable to larger systems [49]. In [49] and [48] an ad-hoc method of handling distributed air traffic management systems with coupled constraints is proposed, making use of model predictive control. They conclude that their approach works better for low traffic densities and that some aircraft fly long periods without updates.

Another approach utilizing decentralized optimization is presented in a paper by Inalhan et al. [50]. They make use of kinematic aircraft models and adhere to separation requirements between all aircraft. Inalhan et al. do not have a centralized agent regulating the coordination scheme. Instead, they use a sequential optimization scheme regulating the separation criteria. The cost function used to drive the vehicles towards their destination penalized the error between the destination and the current position. Pareto optimality can only be guaranteed when local problems have dominating local convexity or when connections are weak.

One example of 4D trajectories using MPC can be found in [9], where Interval Management (IM) is conducted satisfying spacing objectives. In this paper model predictive control is utilized for trajectory optimization while maintaining adequate spacing. MPC for ATC has been used for conflict resolution as well as 4D trajectory optimization. Using constraints to adhere to the minimum required separation distance at all times. Computation times are a limiting factor due to the vast time horizon necessary over large distances.

The trajectory optimization in this paper is performed only in the vertical plane. The method proposed by Weitz and Bai uses MPC to optimize the nominal trajectory in small increments. The optimization did not always yield a feasible solution, during their case studies, for a small planning horizon of less than 5NM, failure rates reach up to 30%. Also, they concluded that for more substantial initial errors, where fast trajectories are required, the NLP convergence is not guaranteed. Furthermore, the computation time required even for the shorter planning horizons is not near real-time and too time consuming for implementation.

In the paper by Bousson [11], a model predictive controller for air traffic separation is given, using a kinematic model closely resembling the model presented in Section 2.3. Bounds are placed on maximum and minimum velocities and the maximum and minimum bank angle and acceleration inputs. The prediction horizon is 120 seconds and the control time steps used in this paper is 10 seconds. The interesting part of this paper is that the set-up of the optimization problem presented by Bousson is, in essence, similar to the centralized MPC proposed in this work. The method in [11], proves that a relatively simple MPC can be used to find a small scale (Bousson used four aircraft in the simulation) feasible solutions.

Due to the computational complexity, real-time applications utilizing a centralized optimization scheme are rarely possible [21]. D'Amato et al. [21] propose to reduce the computational complexity of the positional constraints by using the right of way rules. By using this rule-based approach, the position constraints can be formulated as linear and convex. The problem of immense computation times when adding multiple agents to the system is also expected for the proposed controller. Therefore, the right of way rules might be utilized to reduce the numerical complexity of the optimization problem.

4

Decentralized and Multi-Level MPC Architectures

In this chapter, decentralized, distributed and multi-level control are introduced. Firstly, multi-level controller schemes are introduced in Section 4.1. Secondly, the concept of setpoints and how setpoints can be used in multi-level structures are described in Section 4.2. Followed by a discussion about the differences between decentralized and distributed control schemes in Section 4.3. This chapter is concluded with multi-level distributed or decentralized control implementation examples, in Section 4.4.

4.1. Multi-level Control and Control System Architecture

In the review paper by R. Scattolini [51], an extensive overview of distributed and decentralized hierarchical architectures can be found. A few key concepts and important elements described by Scattolini shall be discussed throughout this Chapter.

When a large scale system is subdivided into multiple levels, the reason is typically to reduce the complexity of the overall system. In a top-down control architecture, the system can have different functionalities and operate on different levels of detail or different levels of abstraction [52].

Hierarchical control with at least two control levels is used most often in two different cases. The first case is when two subsystems or levels of the system have different dynamic behavior, such as fast dynamics and slow dynamics that are hard or inefficient to control with a single controller.

The second case is when a system requires optimization at different update rates for different levels of the subsystems (singularly perturbed systems), this notion can be observed in Figure 4.2 and Figure 4.3.

In the context of MPC, these systems do still lack systematic design methods that guarantee well-assessed properties [51]. According to Scattolini, it is not difficult for an MPC regulator to transmit future control actions and future state trajectories to other neighboring local control units. Nonetheless, there is no systematic method to select the best-distributed control strategy [51].

For ATC applications, a global solution is required where adequate separation for all aircraft is guaranteed at all times. For managing the separation constraints and the global flow rate, a reiteration of the optimization problem at the scale of minutes is sufficient. Whereas for individual aircraft, having a faster update rate of, based on an airlines' preferred cost index, on a scale of seconds might be preferable. The fact that with MPC future control actions and the corresponding state trajectories can be predicted makes it a suitable control method for distributed systems to achieve comparable performances to centralized control systems.

4.2. Setpoint Optimization

The predicted long-term future states can be used by the lower levels as setpoints. In [53], a review is provided of hierarchical control structures utilizing MPC control on the top level of the control structure. In this paper, the temporal decomposition of a control problem through dynamic setpoint optimization is explored. The

MPC or constraint level determines a trajectory of setpoints and constraints, which are used as a reference by the direct control level. Due to the inaccurate nature of future predictions, only the first few setpoints are transmitted to the lower-level controllers. At each level of the controller, the same optimization problem is pursued. Yet, for different time horizons, with different control time steps [53].

The concept of utilizing controllers at different hierarchical levels and using setpoint trajectory optimization will be used and tested in the case studies applied to air traffic control during this research project. A case study where dynamic setpoint optimization for temporal decomposition is also applied is in the paper [5], which will be discussed in Section 4.4.

The centralized controller must be aware of the operational constraints of all the lower levels and must also consider the possibility of disturbances to guarantee the feasibility of the whole system. The lower subsystems must feedback information about their solution to the upper levels such that there is not a significant deviation from what the higher levels demands and the lower levels were capable of achieving [51].

For industrial or process control problems, multi-level control problems have often been applied to achieve global stability. These solutions have been of the first type, where slow and fast dynamics are regulated at different control levels. Often the fast dynamics are regulated by PID controllers while the slower optimization setpoints are determined by MPC controllers.

In Figure 4.1 the optimization structure of a multi-level system is portrayed. On the top level, a plant-wide optimization is performed. This plant wide-optimization result is the input for the local optimizing agents. Only the local optimization is connected to the actual process control and capable of influencing the control inputs (manipulated variables) while steering the controlled variables.

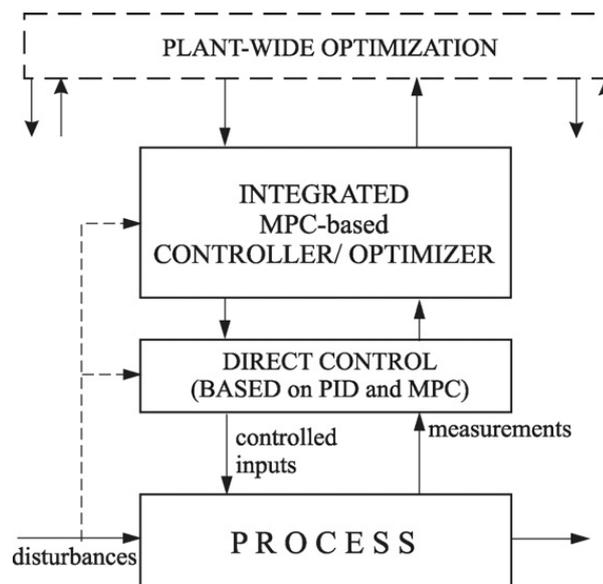


Figure 4.1: A hierarchical control structure [53].

Scattolini describes in his 2009 review paper [51], that MPC can be used at any level of the controller to take full advantage of the input, state and output constraints at each level. Despite this fact, not many applications or research had thus far been conducted where multiple hierarchical MPC controllers have been combined. One paper in which multi-level control is applied for air traffic control is by Pappas et al. in 2000 [52]. They use a less complicated model of the same system at the higher level and a more accurate model on the lower level, with a shorter prediction horizon. Different from this paper will be that both levels apply similar model complexities at different sampling rates, gaining computational efficiency necessary due to large sample times at a higher level.

4.3. Decentralized and Distributed Control

There are many reasons for dividing a control problem into multiple decentralized or distributed control problems. The foremost reasons are that dividing a large scale problem into a smaller scale problem decreases

the mathematical complexity, in terms of computation time and increases overall resilience to failures of the system. When a system is made up of many different subsystems, a centralized optimization problem becomes more complex, with every agent added to the set.

The main difference between decentralized and distributed control originates in the way information of other agents in the system is applied to achieve the control goals. Within decentralized control systems, the knowledge that different agents have of each other is not used to find the optimal solution cooperatively. Whereas in distributed control systems, agents do share information about their actions and intentions, to reach a global solution cooperatively. They take the information from their neighbors into account when determining their control actions. Some level of cooperation or interaction between different agents is, in most cases, beneficial in terms of the pursuit of the global optimal solution. There is, however, an extra level of difficulty that needs to be handled in the case of distributed systems. When separation constraints are introduced in a decentralized system and the decentralized agents are coupled through the constraints, collision avoidance cannot be guaranteed for decentralized RHC structures [54].

The interconnection structure cannot be chosen to be full because that would scale up the optimization problem, and the computation speed advantage of a decentralized optimization approach is reduced. Also, due to the discrepancy between predicted and actual neighbor trajectories, the local problems are not guaranteed to be feasible [54]. In the paper by Keviczky et al. [54], an emergency controller is proposed that brings the velocity of the controlled system to zero in case an infeasible local problem is detected. For the application of passenger planes, this is not feasible, since there is a minimum speed required to sustain flight.

The extra challenge that comes with distributed controllers is how the different controllers are synchronized and how the order of information flow is managed. Since each agent in the system cooperates to find the best solution and they rely on each other's decisions. One agent control actions might change based on the intents of another agent. A communication flow problem, much resembling the 'chicken and egg' problem emerges.

Luckily, there have been multiple studies into distributed optimization where this problem is investigated. Such as the study conducted by Negenborn et al. [55] where serial and parallel optimization schemes are compared. In this paper, it is concluded that parallel optimization is not necessarily better in terms of computation speed and finding the optimal solution. Serial optimization even has preferable features in their experiments.

In the paper by Chaloulos et al. [8], a distributed model predictive control approach is used to optimize the path for three crossing aircraft. They discovered that the order of the optimization sequence can have a substantial effect on the outcome. Different results were obtained when the optimization was performed in a sequential 'round-robin' fashion, or in random order. They propose to solve this problem by reducing the optimization space for the remaining aircraft, once one aircraft has determined its control action. A cooperative terminal cost function was introduced to couple the decentralized agents. Each aircraft still optimizes its optimal path, however, a fairness factor is introduced in the cost function to take into account the effect of the solution that an aircraft has on the other aircraft in the system. The solution proposed in the paper does have the risk of becoming computationally challenging once more aircraft enter the system and even though the optimization keeps other agents into account, the optimization is still performed in a sequential matter. Chaloulos et al. use a time step of 5 minutes and a prediction horizon of $N = 4$ for their higher-level MPC to perform far ahead conflict-resolution.

The information flow problem described above is only apparent when individual agents cooperatively aim to achieve a global optimum. When a system is fully decentralized and agents do not rely on each other's decisions, the issue of intent-based information streams is non-existent. However, for the highly coupled air traffic control field where safety is the priority, some level of cooperation is required. The global optimization and safety constraints can be moved to another hierarchical control level to move the issue of sequential optimization from the individual agent level to a higher control level. This idea is much like the traditional way in which air traffic is currently managed. Where a single human controller manages multiple aircraft at once and tries to achieve an optimal solution for the whole system while adhering to all safety constraints. In classical air traffic control, there is also no need for individual aircraft to communicate and cooperate, except for emergencies when ASAS is applied.

A bi-level control architecture could take on this task and safely optimize air traffic in a way that feels familiar to current practice. In such a system, the individual aircraft do not have to interact with each other,

since the centralized controller handles the interaction between the systems in terms of tactical and strategic separation assistance. The decentralized controlled aircraft transmit their state information to a centralized controller, which combines the data to perform a global optimization.

A system whereby the decentralized agents also communicate with their neighbors can be part of future research. The ADMM (alternating direction method of multipliers) could prove to be a valuable method to handle coupled constraints [4][7]. The problem statement would then shift the lower-level controller from decentralized control to distributed control. This method could be implemented during future research.

4.4. Applications of Multi-Level Controllers

The main challenge of designing hierarchical system architecture is to determine the functionality and objective for each level of the system [52]. The careful conceptual design of the different control levels of the system is a fundamental step when designing a multi-level system. Therefore, the concept and tasks of each level have to be determined, when this notion is related to the development of the multi-level controller of this thesis.

In a 1998 paper from Ying et al. [56], a two-stage control scheme was investigated and compared to a single-stage controller in terms of stability and optimization result. The controller investigated in this research was a single QDMC (Quadratic Dynamic Matrix Controller) and a QDMC with an additional MPC stage. The single-stage QDMC had a steady-state offset and did not take into account the economic objectives. The second stage resolved the steady-state issues and addressed the economic cost function of the whole plant.

In this research, the stability of such a multi-stage system, as well as its ability to achieve convergence towards a reference, is investigated. In this chemical process optimization, a steady-state condition is achieved by performing an economic optimization on the higher optimization level, determining setpoints necessary which the sub-levels have to achieve. For their applications, the two-level system outperforms the single level system in terms of stability as well as cost function optimization.

Another application where hierarchical model predictive control was applied is described in the study by M. Brdys et al. [5]. In this paper, a wastewater treatment system is controlled using three separate hierarchical levels. The Supervisory level, the Optimizing level, and the Follow-Up control level. The top level is the supervisory control level and information from all the control levels is available to the supervisory controller. The supervisory level determines the control strategy and the coordination of the plant. The optimizing control level defines the setpoints of the plant and the follow-up control levels achieve these setpoints by means of classical PID control.

The optimizing control level is further separated into three control sub-levels which operate at different time scales. One level for the slow dynamics, the medium dynamics and one for the fast dynamics of the plant. The reason that the researchers choose to split the optimization level into three separate levels was that using a single controller to handle all the dynamics would lead to a complex inefficient controller. The short control time steps required for the fast dynamics combined with the large prediction horizon for the slow dynamics would yield a high dimensional, uncertain optimization problem that would lead to computational complications for real-time applications [5].

The SCL (slow control sublevel) operates on one day control steps over a horizon of weeks or months. The MCL (medium control sublevel) operates on one hour control intervals over a horizon of a day. The fast control sublevel (FCL) uses one minute control steps and predicts one hour forward in time. A representation of what this looks like can be seen in Figure 4.3 and Figure 4.2. An analogy can be drawn to air traffic control here, where, vast prediction horizons are needed to control the air traffic flow and manage separation distances over broad time horizons for thousands of nautical miles. However, fast inputs are required to steer and control the aircraft optimally on a short time horizon.

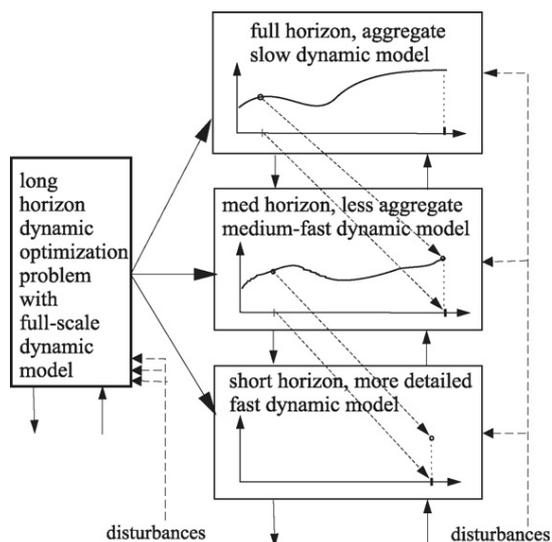


Figure 4.2: Decomposition of different dynamic rates within the system [53].

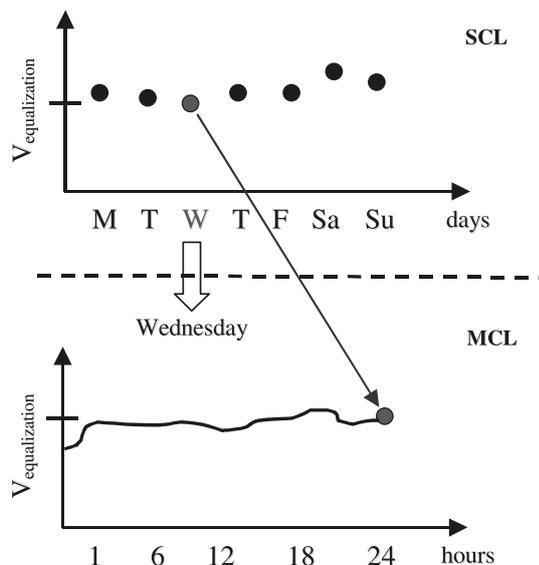


Figure 4.3: Different update rates at different levels of the controller [5].

For the wastewater-treatment plant study, no comparison is made between a single level or multi-level controller to study the controller performance relative to a single centralized controller. This paper only describes the working principle of the controller and the motivation for its development. The results are promising, and the controller successfully controls the plant. Studying the effects of a multi-level controller for air traffic applications might yield exciting results.

5

Bi-Level Model Predictive Controller Conceptual Design

In this chapter, multiple potential bi-level model predictive controllers concepts for air traffic control applications are presented. First, the bi-level MPC controller and the intended objectives are introduced in Section 5.1, the benefits of multiple control levels are described here, as well as, the motivation for the chosen approach. In Section 5.2, a recap of several important notions from relevant literature is discussed, that will be used in particular during the development of the controller. This is followed by Section 5.3, where practical considerations are mentioned to form the bridge between the theoretical controller concept and the real-life implementation. In Section 5.4, four controllers are discussed on a conceptual level. Finally, in Section 5.5, the integration of the decentralized and centralized controller is discussed and the formulation of the optimization problem is presented.

5.1. Bi-level Controller Working Principles

The controller proposed in this research project is a combination of a centralized MPC and a decentralized MPC that cooperate by operating on two different hierarchical levels with different control sample times and prediction horizons. The centralized controller predicts further into the future than the decentralized controllers, that have shorter time horizons.

To reduce the computation complexity of the centralized controller, the control sample time is adjusted such that the centralized controller uses coarser intermediate steps, providing a rougher solution. To regain the lost accuracy, a decentralized control level is introduced, using a faster control sample time. Figure 5.1 gives an indication of the optimization control sample time over time for both control hierarchies and the resulting level of detail of the solution.

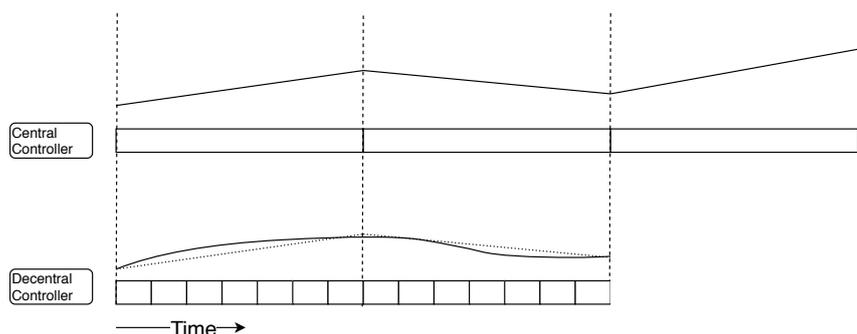


Figure 5.1: Difference in optimization control sample time and the resulting state trajectories.

Both controllers execute a similar control task, whereby each aircraft is guided towards their destination. For each aircraft, two control inputs are determined, an acceleration input and a bank angle setting (implying a certain turn radius).

The centralized controller has an additional task, to separate global air traffic by maintaining an adequate distance between each aircraft. The decentralized controllers are not aware of the presence of other aircraft and must follow the conflict-free trajectories determined by the centralized controller, despite those trajectories having a reduced level of detail. Consequently, a margin can be provided to the centralized solutions, such that the decentralized controllers are granted autonomy to find a more optimal solution within the given space.

Since the outputs of the centralized controller are acceleration and bank angle, it is possible to analytically determine the corresponding future positions, using the aircraft model and the optimized centralized controller input sequence. The discretized points at each control sample time in the prediction horizon, form the centralized controllers' trajectory. These points are hereafter called setpoints (SP) and will be converted to dynamic constraints that are provided to the decentralized controllers.

The illustration in Figure 5.2, provides a schematic representation of this concept. The centralized controller avoids a forbidden area and determines a rough conflict free trajectory for two different aircraft towards their destination. The red lines in Figure 5.2, represent the separation constraints that the centralized controller must obey.

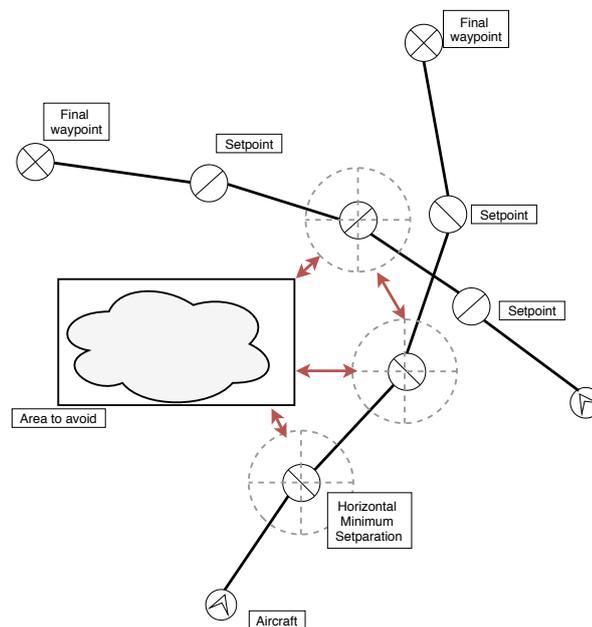


Figure 5.2: Centralized conflict-free trajectories determination using setpoint while avoiding forbidden area.

5.2. Concept Motivation From Literature

Similarly to most next-generation air traffic control concepts, the control system proposed in this chapter operates on a 2D horizontal plane. Vertical conflict resolution is initially not considered for three different reasons, horizontal 2D trajectories are easier to visualize than 3D trajectories, heading changes result in more passenger comfort than flight level changes [37], and heading changes are also more fuel-efficient than altitude changes [37].

Apart from heading changes, flight speed adjustments are also controlled by the bi-level controller. Speed adjustments require to be applied over vast distances to have a noteworthy effect. For this reason, speed adjustments are not commonly used by air traffic controllers. However, when using an automated system with a large look-ahead time, conflicts can be predicted and solved at an early stage with minor speed adjustments.

Subliminal speed control has become more promising with the introduction of accurate trajectory control methods [37]. In [57], Rey et al. evaluate a method that is limited to subliminal speed control for conflict deconfliction, they prove that speed control can be an important attribute in solving conflicting situations and reducing air traffic controller workload. They use a linear approximation and provide a MILP solution that provides a global optimum in a few seconds.

Speed control will, therefore, be implemented in conjunction with heading control to improve the performance of the bi-level MPC controller.

5.3. Practical Consideration and Link to Reality

In classical air traffic control, the management system is divided into separate levels, operating on different time scales. These levels include the tactical flow management, separation management, and ASAS systems. Tactical conflict resolution is currently performed at a prediction horizon or looks ahead time of approximately 20 minutes and separation management is handled at the time scale of 5-7 minutes [45], see Figure 5.3. A resemblance of such a leveled structure is a foundation for the proposed controller, due to the temporal decomposition of the two controller levels.

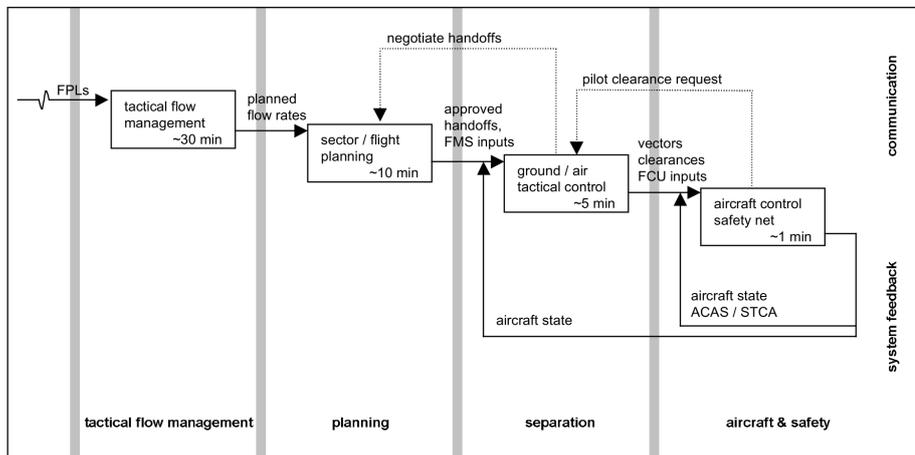


Figure 5.3: Layered structure in air traffic control [45].

In a realistic scenario, the centralized controller (air traffic control center) has a less accurate knowledge of the aircraft parameters (weight, aerodynamic characteristics, etc.) and wind velocities than the pilot. To create an analogy with this realistic scenario, while running the experiments, the centralized and decentralized controllers use different values for the drag polar and the aircraft mass.

The model parameters used by the decentralized controller are assumed to be more accurate, which enables the decentralized controller to determine a more accurate optimization. Mass is assumed to be constant during the simulations, even though it could be included based on the fuel flow, which can be part of future research.

To simulate a more realistic scenario and to test the response to inaccuracies, disturbances can be added to the system by means of introducing a wind field. The introduction of wind to the system is likely to highlight the beneficial effect of having two levels of the control system. The centralized controller has less knowledge of the wind field in comparison to the decentralized controllers.

Another possibility that emerges due to the hierarchical control structure is that each aircraft can optimize a cost function with different weights for fuel consumption and flight time. This creates another analogy with the true situation where every airliner can determine their preferred cost index, based on their business model. This allows for a different balance between flight time and fuel flow for each aircraft. The cost index that a certain aircraft operates at is unknown to the centralized controller. The centralized controller uses the same, global, cost index for all aircraft, which is likely to be different from that of the decentralized aircraft.

5.4. Controller Integration and Conceptual Development

In this section, four controller integration concepts are given, as well as, an introduction of the practicalities concerning the real-time operation of the bi-level controller.

5.4.1. Concept 1: Heading and Velocity Constraints

Figures (5.4-5.7), show four different conceptual solutions for converting the setpoint to constraints for the decentralized controllers to use for their short term problems. The first concept can be found in Figure 5.4. In this concept, constraints on the heading are applied such that the setpoint falls within these bounds. Furthermore, a minimum velocity constraint is applied, making sure the aircraft arrives in the green area within

a certain time. The maximum and minimum velocity constraints are meant to limit the longitudinal position where the aircraft is allowed to be. In this way, the horizontal position of the aircraft is fully constrained. The decentralized controllers are free to optimize their path as long as the positional constraints are adhered to.

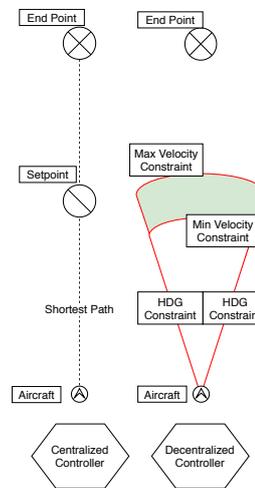


Figure 5.4: Concept 1, heading and velocity constraints.

5.4.2. Concept 2: Longitudinal and Velocity Constraints

The second concept achieves the same results as the first concept, however, instead of applying a constraint on the heading, the longitudinal constraint is placed in terms of maximum and minimum x coordinates. It can be seen in Figure 5.5, what this would look like when flying at a heading of 0 degrees on the left side. However, it must be noted that an axis rotation is required when the angle is not 0 degrees. Instead of using the Earth-Fixed Reference Frame, the Body-Fixed Reference Frame must be used. Before the x bounds are applied, the axis are rotated with the initial heading of the aircraft, to be able to set the x bounds with respect to the aircraft. The velocity constraints are applied in the same manner as in concept 1.

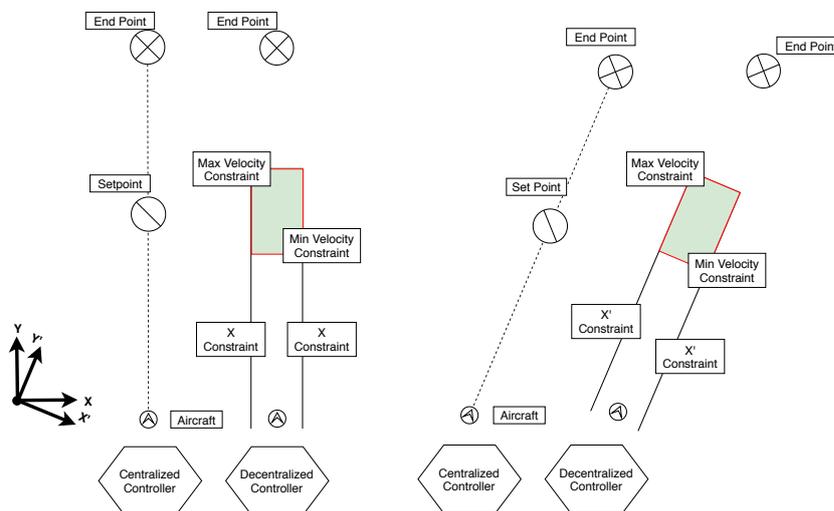


Figure 5.5: Concept 2, velocity and longitudinal constraint with axis transformation

5.4.3. Concept 3: Combination of Concept 1 and Concept 2

The third concept is shown in Figure 5.6 along with a time propagation. The third concept is a combination of the former two concepts. Where the triangular constraint shape of Figure 5.4 coincides with the squared shape of concept 2 (Figure 5.5). The first section of the feasible area is partially constrained by the physical

limitations of the maximum turn radius of the aircraft. The longitudinal constraint keeps the aircraft relatively close to the shortest path. The rotation of the axis is again required to place the longitudinal constraints conveniently. It is important to note that the centralized controller can give updated constraints before the aircraft reaches the green area.

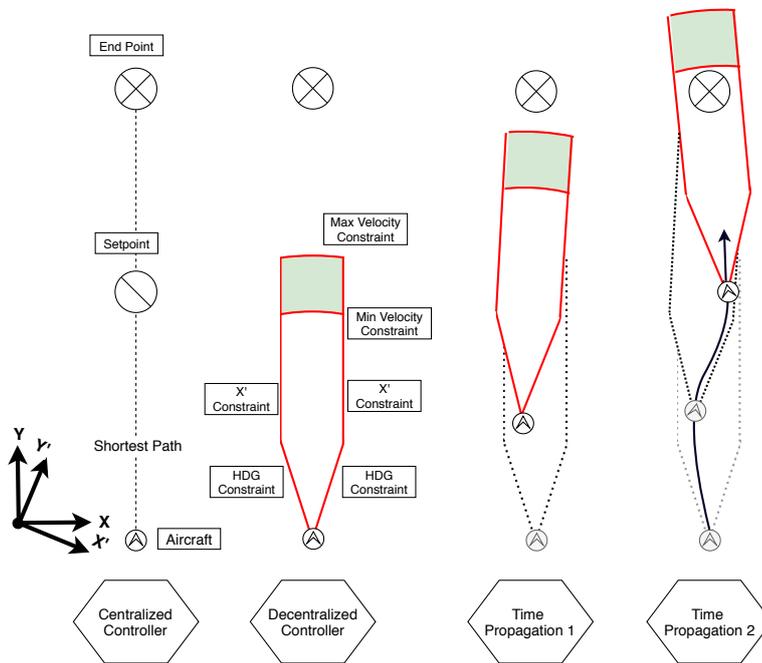


Figure 5.6: Concept 3, combined longitudinal and heading constraints with velocity constraints.

5.4.4. Concept 4: Maximum Position Constraint Relative to Setpoints

The final concept only uses positional constraint to ensure that the decentralized controllers stay close to the central setpoints, and is illustrated in Figure 5.7.

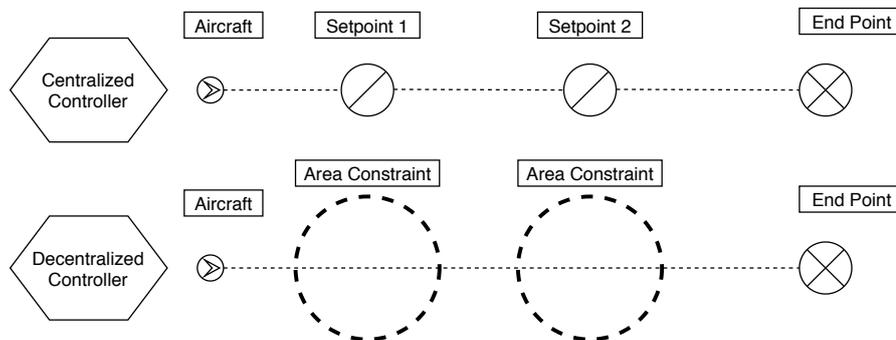


Figure 5.7: Concept 4, setpoints to area constraints conversion.

The setpoints are converted to circular area's and the decentralized controller must be within a certain radius at the corresponding moment in time.

The heading and velocity constraints are implicitly applied by use of the positions constraint but can be applied explicitly as well, to define a different area shape.

5.4.5. Real-Time Simulation Consideration

The systematic representation in Figure 5.8, describes the time wise implementation of the two controllers. The decentralized controller starts after the centralized controller optimizes the trajectories for all aircraft in the set. The measured states are fed back to the centralized controller for the next optimization cycle.

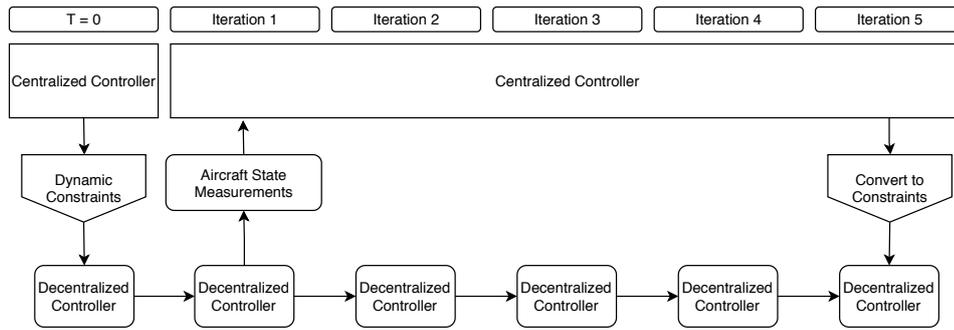


Figure 5.8: Schematic representation of the update rate of the decentralized and centralized controllers.

The centralized optimization at the next time step induces a time delay and causes new constraints for the decentralized controllers, to be available several iterations later. This causes 2 problems, the decentralized controller performs a number of iterations without receiving new constraints and when the centralized optimization provides the new constraints they are based on old measurement.

Figure 5.9, shows what the schematic block diagram from Figure 5.8 entails in practice. In this example, the centralized controller can determine new setpoints every two iterations. The aircraft at $T = 2$ obtains the constraints calculated at $T = 1$ and the aircraft at $T = 4$ receives the constraints based on measurement at $T = 3$.

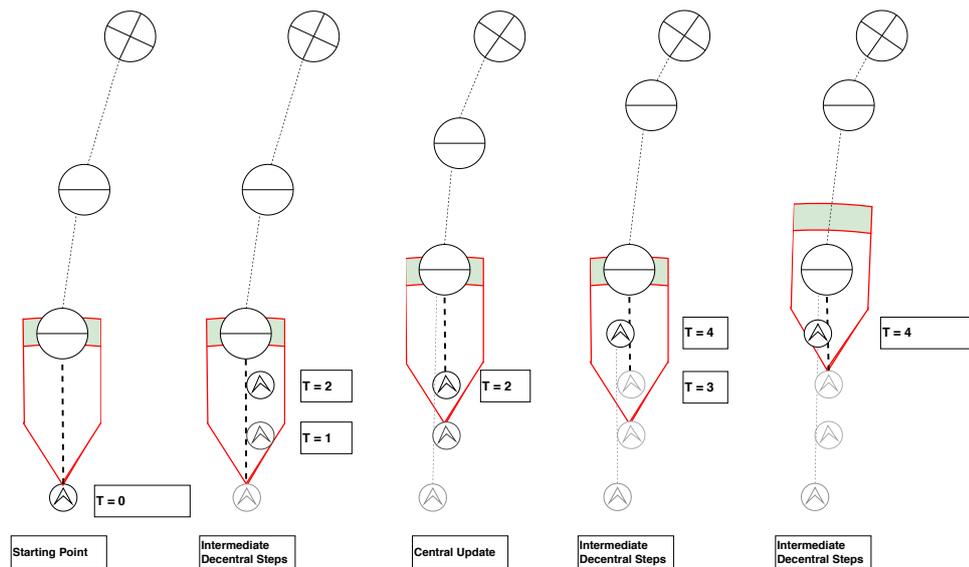


Figure 5.9: Propagation of an aircraft over time with intermediate constraint updates by the centralized controller.

The practical implications of running the systems in a real-time application do not fall within the scope of this project. For this project, the assumption is made that the centralized controller operates in sequence and always uses the most recent state estimations. Adapting the bi-level controller to have the centralized controller work in parallel, is a recommendation for future research.

5.4.6. Extension of the Decentralized Prediction Horizon

The decentralized controllers' prediction horizons can be extended to include setpoints further ahead into the future, such that the constraints span multiple setpoints, possibly improving the performance of the system. Figure 5.10 shows a visual interpretation of what this entails. Two examples are shown, for the first example (left side) the setpoints both lie on the shortest path to the final waypoint. The second example (right side) shows what might happen when two setpoints form an angle.

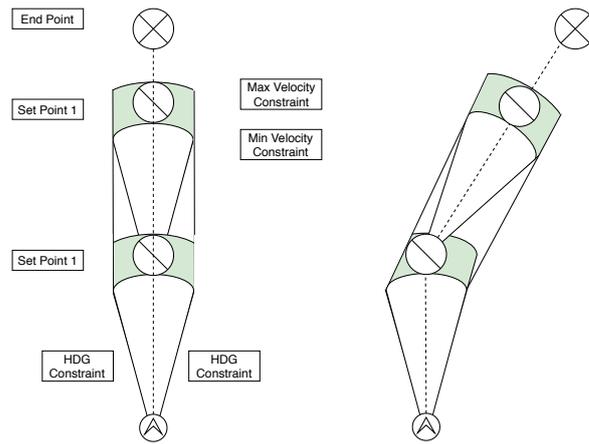


Figure 5.10: Increase in decentralized prediction horizon to include multiple setpoints.

When a large angle is formed between two setpoints, the importance of having at least 2 setpoints within the prediction horizon of the decentralized controllers, becomes apparent. If the second setpoint is not included, the aircraft might arrive too far to the left side of the first setpoint (Figure 5.10), forcing it to make large control efforts to re-adjust its path to steer towards the next setpoint.

5.4.7. Determination of the Constraint Boundaries

The constraint boundary sizes can be kept constant throughout the entire simulation or the boundary size can be adjusted during the simulation. Also, the second and third setpoints can work with different boundaries than the first setpoint. The effect of changing setpoint sizes needs to be explored in the future.

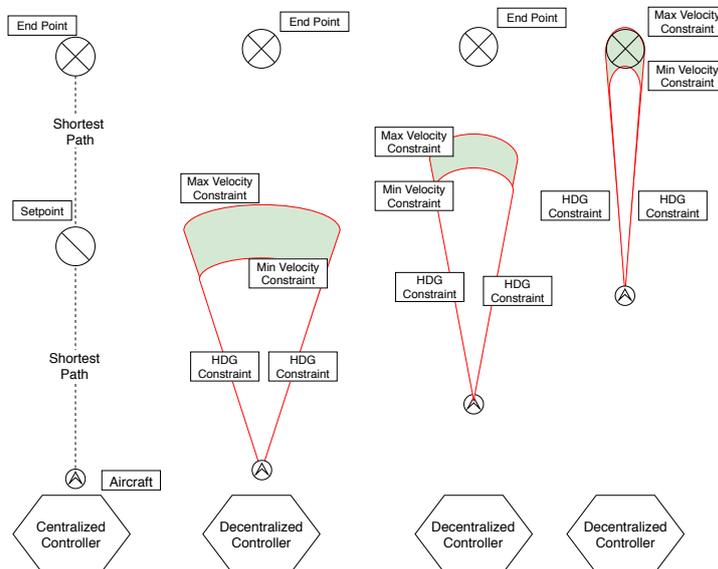


Figure 5.11: Decrease in constraint sizes when distance to destination decreases.

To make sure that the aircraft arrives close enough to the destination waypoint, the boundaries must become tighter near the end of the flight path, see Figure 5.11. The boundaries can be reduced in such that the aircraft arrives close to the final destination. This can be implemented by making the size of the boundaries proportional to the remaining distance to the final waypoint.

5.5. Bi-Level Model Predictive Control Formulation

The decentralized controllers determine the optimal control input sequence for every aircraft. In reality, the acceleration and bank angle are calculated for every aircraft on separate computers. In this practical approach, the independent aircraft are optimized sequentially, on the same computer. Once a control input is determined for every aircraft, the simulation advances such that aircraft receive control inputs before the simulation continues. Since the aircraft on the decentralized level are not coupled the sequential optimization yields the same result as a parallel optimizing technique. When the aircraft is coupled by the separation constraint on the lower level for conflict resolution as well, the solution changes by taking the order of optimization into account.

In [55], it is concluded that the parallel computation scheme is not necessarily better and sequential optimization schemes are even preferable in terms of computation time and solution quality. However, in [8] it is observed that the order of optimization significantly affects the solution outcome. This is a topic for future research.

The controller block diagram showing the systematic design of the bi-level controller and the interconnections between the centralized and (one) decentralized controller is presented in Figure 5.12. The formulation of the constraints and cost functions is presented in the remainder of this Section.

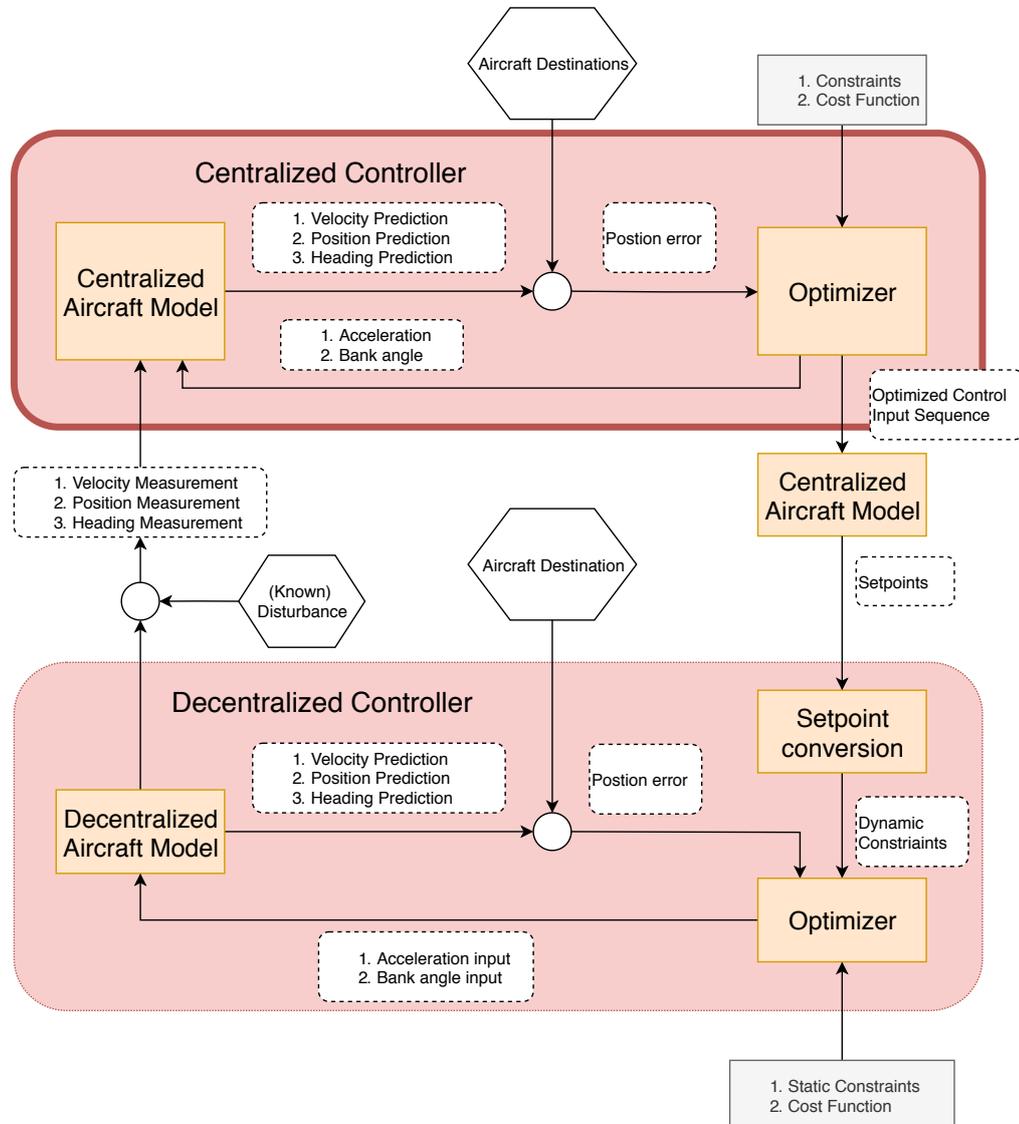


Figure 5.12: Bi-level controller control diagram.

5.5.1. Centralized and Decentralized Cost Functions

The cost function is given by Equation 5.1, and includes a term for total flight time, fuel consumption, the distance between the destination and the current coordinates and the change in control input.

$$\begin{aligned}
J(k) = & W_1 \cdot \sum_{i=1}^{H_p} \sqrt{(y(k+i|k) - y_{\text{final}})^2 + (x(k+i|k) - x_{\text{final}})^2} \\
& + W_2 \cdot \sum_{i=1}^{H_p} \text{Fuel Flow}(k+i|k) \\
& + W_3 \cdot \sum_{i=1}^{H_u-1} \sqrt{(u(k+i|k) - u(k+i-1|k))^2}
\end{aligned} \tag{5.1}$$

where, W_1 represents the weight assigned to distance between the origin and the final destination and W_2 is the weight associated with the total fuel consumption. The weight given to the difference between two consecutive control inputs is given by the value W_3 .

5.5.2. Centralized and Decentralized Constraints

In this section, the centralized and decentralized velocity constraints and forbidden area constraints are formulated. These constraints apply to both levels of the controller. The velocity of each aircraft is constrained according to:

$$V_{\min} - 0.94 \cdot V_0 > 0 \tag{5.2}$$

and,

$$V_{\max} - 1.03 \cdot V_0 > 0 \tag{5.3}$$

the acceleration input is constrained between,

$$-0.5 \frac{m}{s^2} < a < 0.5 \frac{m}{s^2} \tag{5.4}$$

the minimum and maximum bank angle input,

$$-25^\circ < \psi < 25^\circ \tag{5.5}$$

Finally, another constraint that applies to both control levels is the forbidden area constraint. This constraint describes a forbidden circular area such that an aircraft stays clear of a coordinate set with a certain radius,

$$\sqrt{\Delta x_{oj}^2 + \Delta y_{oj}^2} - \delta_{\text{MinSep}} > 0 \tag{5.6}$$

5.5.3. Centralized Separation Constraint

Figure 5.13 shows possible conflicts when the setpoints lie too far apart. Even though the respective setpoints have adequate distance between them, aircraft 1 and aircraft 2 are in conflict between setpoints 1 and 2. One solution to solve this issue is not to allow lines in between 2 setpoints to cross.

Another method to avoid this issue is to make the separation constraints larger or make setpoints lie closer together. If the next setpoint and the initial position of the aircraft fall within the separation boundaries, the unwanted situation, shown in Figure 5.13, can not occur. Figure 5.14 shows an infeasible situation because the setpoints lie within each other's separation margins. Increasing the size of the separation margin is unwanted since that would reduce the effective airspace that could be used. The first solution of not allowing paths to cross in between setpoints is preferable because that allows for setpoints to lie further apart.

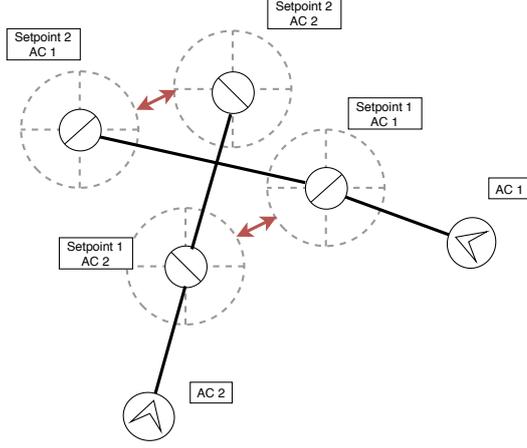


Figure 5.13: Possible conflicts even though separation constraints on setpoints are obeyed when the setpoints are too far apart or the separation distance is too small.

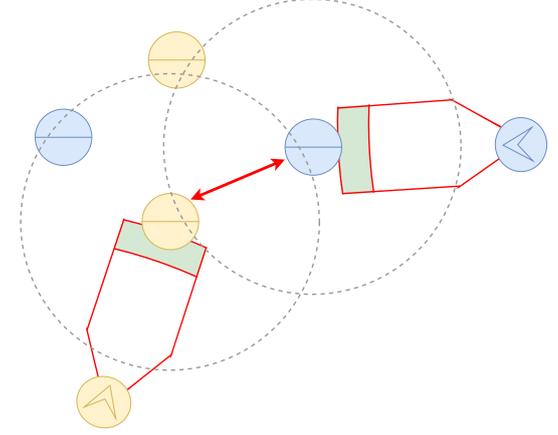


Figure 5.14: Bringing setpoints closer together solves this problem, by making this solution infeasible.

A general convention is that the lateral separation distance between aircraft should be at least 5 nautical miles (or 9260 meters). If the average cruise speed of a passenger jet is 250 m/s, loss of separation can occur in approximately 35 seconds. So the control sample time of the centralized controller must be much higher than 35 seconds. When two aircraft are in a potential head-on conflict, this margin is halved.

Since the centralized sample time is likely to be longer, an aircraft separation constraint must be implemented. The separation constraint is based on the relative position, heading and velocity between two aircraft, and is based on the Modified Voltage Potential [33] equations used to determine the closest time of approach and closest point of approach. The separation constraint is given by Equation 5.14, and is obtained by using Equation (5.5.3-5.13)

$$V_{\text{rel}} = V_i - V_j \quad (5.7)$$

$$D_{\text{rel},x} = x_i - x_j \quad (5.8)$$

$$D_{\text{rel},y} = y_i - y_j \quad (5.9)$$

$$V_{\text{rel},x} = V \cos(\varphi) \quad (5.10)$$

$$V_{\text{rel},y} = V \sin(\varphi) \quad (5.11)$$

$$t_{\text{cpa}} = \frac{V_{\text{rel},x} \cdot D_{\text{rel},x} + V_{\text{rel},y} \cdot D_{\text{rel},y}}{V_{\text{rel},x}^2 + V_{\text{rel},y}^2} \quad (5.12)$$

$$D_{\text{cpa}} = D_{\text{rel}} - t_{\text{cpa}} V_{\text{rel}} \quad (5.13)$$

The separation constraint is formulated as,

$$|D_{\text{cpa}}| - \delta_{\text{MaxSep}} > 0 \quad (5.14)$$

5.5.4. Decentralized Maximum Position From Setpoints Constraints

The method of setpoint to constraint conversion is formulated in a way that is analogous and opposite to the forbidden area constraint as described in Section 5.5.2. The decentralized controller is forced to steer the aircraft within the circular proximity of the centralized setpoints.

$$\delta_{\text{MaxSep}} - \sqrt{\Delta x_{s_j}^2 + \Delta y_{s_j}^2} > 0 \quad (5.15)$$

Since the setpoints are determined at larger discrete time steps. This constraint cannot be applied to each iteration in the decentralized prediction horizon. Instead, the constraint is implemented only at the decentralized iteration at the same instance in time (k).

$$\delta_{\text{MaxSep}} - \sqrt{\left(x_{\text{set}}[n_{\text{set}}] - x \left[n_{\text{set}} \frac{\Delta t_c}{\Delta t_d} - k\right]\right)^2 + \left(y_{\text{set}}[n_{\text{set}}] - y \left[n_{\text{set}} \frac{\Delta t_c}{\Delta t_d} - k\right]\right)^2} > 0 \quad (5.16)$$

where n_{set} is an integer value indicating if the constraint applies to setpoint 1, 2, ..., n

6

Thesis Project Timeline and Controller Development Plan

In this chapter, the development phases of the bi-level controller are presented, as well as the projected project timeline. The development of the controller is set up in 3 different phases. In Section 6.1, a global project planning can be found. The preliminary development phase is the initial development of both control levels and explained in detail in Section 6.2. The main development phase is presented in Section 6.3. Finally, during the third phase, the controllers shall be expanded such that multiple agent testing can be performed in BlueSky, this process is explained in Section 6.4.

6.1. Project Planning and Gantt Chart

The experimentation phases of the controller are conducted concurrently with the development phases, in Figure 6.1 the timeline can be seen. The initial development and experiment phases will be included in the preliminary phase of the thesis, the main development and testing phases will be part of the final results of this research project. An extensive Gantt chart can be found in Appendix B.

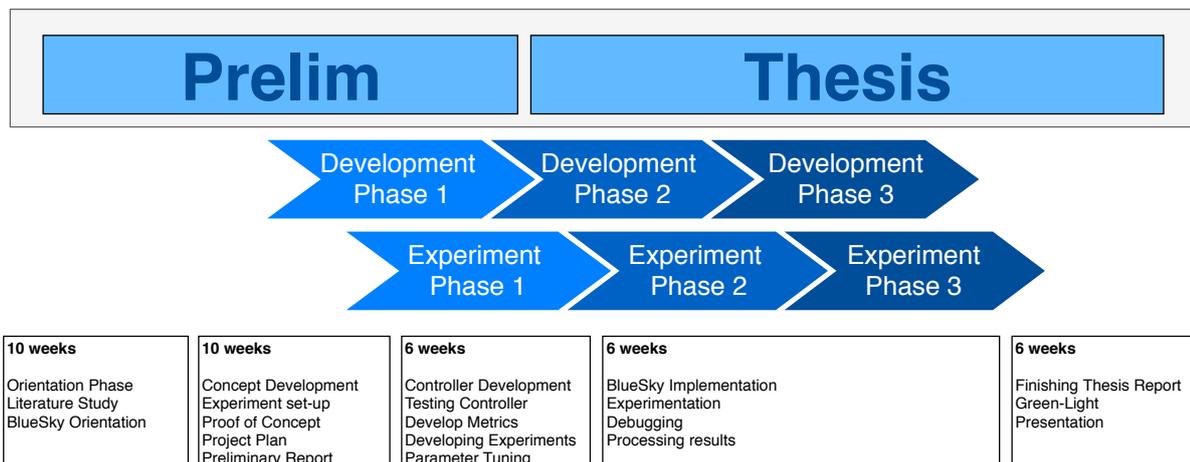


Figure 6.1: Global thesis timeline and project tasks.

Figure 6.2, shows the most important development milestones for the bi-level model predictive controller. Milestones 1.0, 2.0 and 3.0 are part of development Phase I. The centralized and decentralized integration is the most important milestone of the main development phase, Phase II. Finally, the multi-agent and full BlueSky implementation will be part of the third development phase. In Sections (6.2-6.4), these milestones are broken down into the contributing development steps.

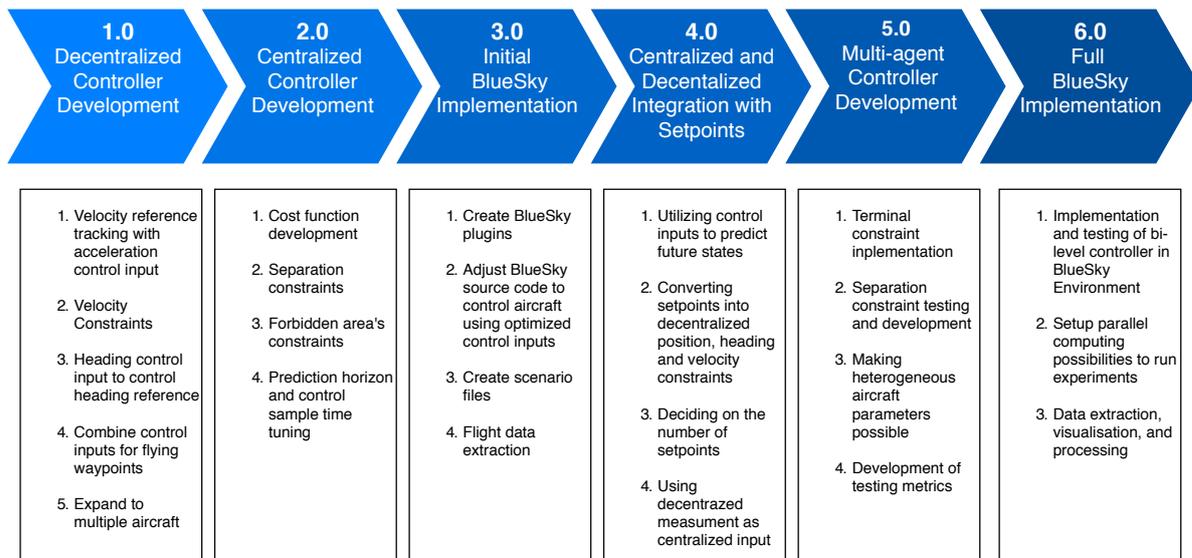


Figure 6.2: Main development milestones.

6.2. Development Phase I - Preliminary Development Stage

During the first phase, an initial version of the centralized and decentralized controllers are developed and tested. The development steps that are taken during the preliminary development phase are presented in chronological order, in Figure 6.3.

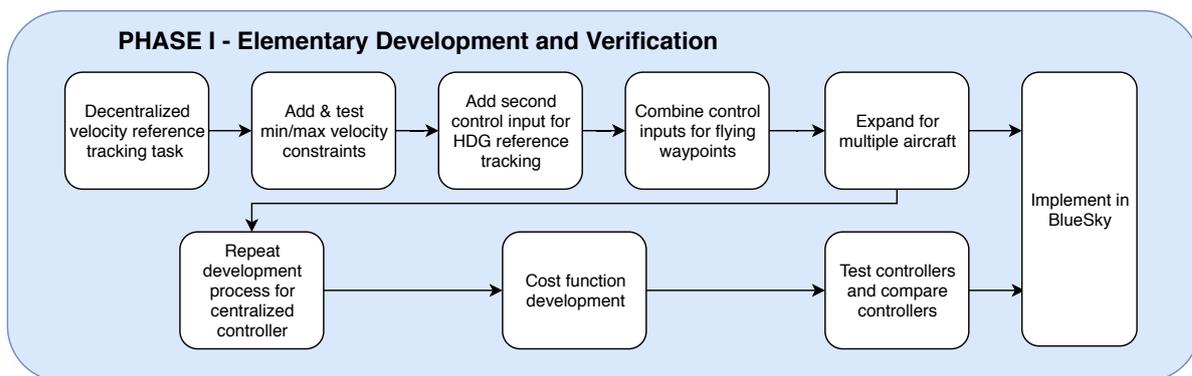


Figure 6.3: Development flowchart for Phase I.

The decentralized controller is developed in Python using the Scipy minimization package for the optimization part of the MPC. The optimization method used is Sequential Least-Squares Programming (SLSQP). This method is chosen because it is one of the few that allows for constraints to be added to the optimization function.

Velocity reference tracking is initially used to verify the performance of the controller. However, a reference on any of the other states, as well as a combination of multiple references, can be used. Several tests are conducted to see if the controller performs as required. The bank angle setting is added as a second control input, allowing the aircraft to change the heading and fly towards their destination.

The development of the centralized controller is started after the decentralized controller is completed and works for the control of multiple aircraft. Instead of performing the optimization sequentially, the entire optimization is vectorized and performed once at every control time step. Engine fuel flow is added as an additional term to the cost function. This is an essential feature for the final controller, where flight time and fuel flow are the optimization parameters. Additionally, it serves as another method to test the cost function. When a relatively high weight is applied to the fuel flow terms, the velocity of the aircraft is expected to decrease to the minimum velocity, resulting in the minimum amount of fuel usage.

Both controllers have to be converted to operational plugins in BlueSky. This is done to evaluate how BlueSky simulations with the bi-level controller are to be implemented.

6.3. Development Phase II - Controller Development and Bi-level Integration

The second development phase focuses on the detailed design of the bi-level controller concept and the combination of the centralized and decentralized controller into a single controller. The flow chart for Phase II can be found in Figure 6.4.

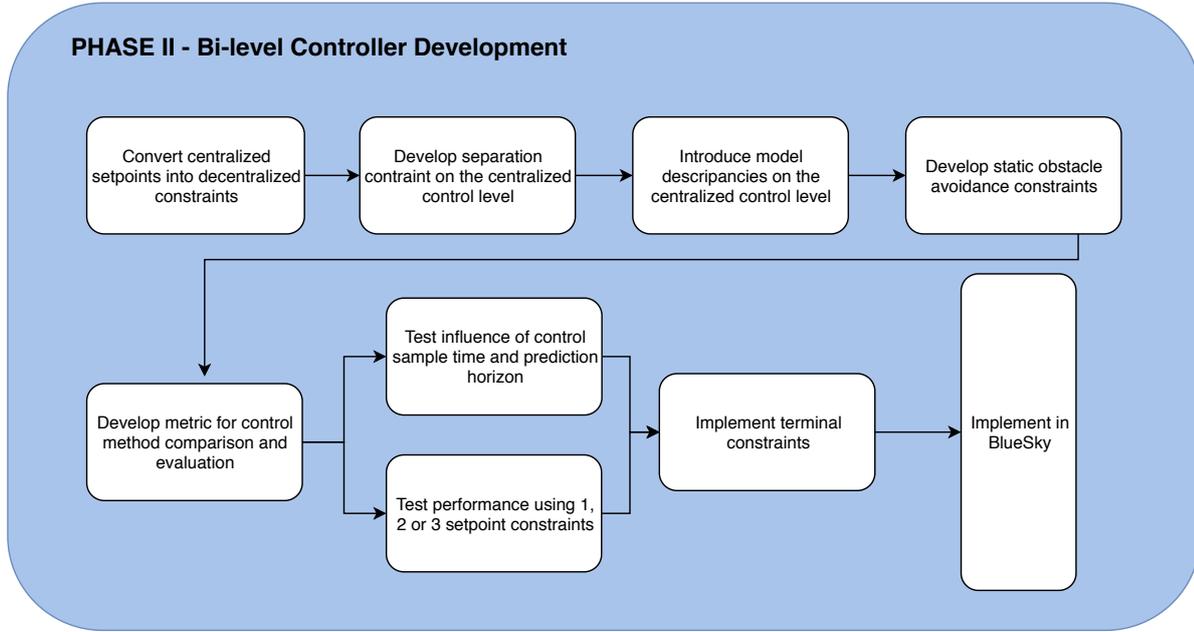


Figure 6.4: Development flowchart for Phase II.

The optimized control inputs computed by the centralized controller have to be converted to setpoints. This is done by using the aircraft model to determine the corresponding predicted coordinates given the optimized control. The predicted aircraft positions, provided by the centralized setpoints can be used to generate positional constraints for the decentralized controllers. The maximum allowed Euclidean distance from each setpoint serves as a constraint for the decentralized controllers.

When the coordinates of the setpoints are available, the shortest path heading can be determined. The heading corresponding to the shortest path is required for all three controller concepts because the heading and longitudinal constraint boundaries are determined with respect to this heading value. The shortest path heading can be calculated using,

$$\varphi_{\text{shortest}} = \arctan\left(\frac{y_{\text{set}} - y}{x_{\text{set}} - x}\right) \quad (6.1)$$

The average velocity between two setpoints is necessary to determine the velocity constraints for the decentralized controller. The average speed is computed using,

$$V_{\text{avg}} = V_0 + \frac{1}{2} a \cdot \Delta t \quad (6.2)$$

Furthermore the separation constraint is further developed as well as the forbidden area constraint so that dynamic and static obstacles can be avoided. The development of a performance evaluation metric is developed. Finally, terminal constraints must be implemented, to have the aircraft decrease its velocity to zero when reaching the destination.

6.4. Development Phase III - Bi-level Controller Implementation in BlueSky

The development of the controller is concluded in Phase III, where the developed system is converted and implemented in BlueSky. The final development steps are presented in Figure 6.5.

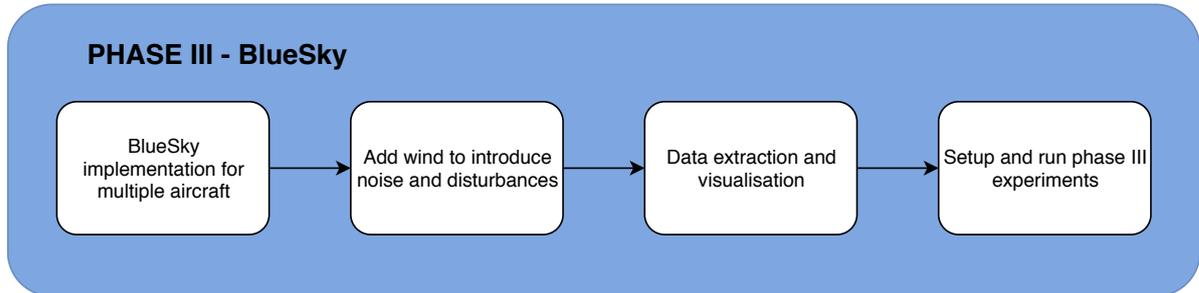


Figure 6.5: Development flowchart for Phase III.

The aircraft model uses measurements of the aircraft position x and y in meters. Whereas, BlueSky gives the aircraft position in longitude and latitude. Therefore, a conversion method must be implemented to be able to use BlueSky longitude and latitude measurements. The wind field can be enabled in BlueSky to generate disturbances. When the controller works for multiple aircraft in BlueSky, the final experiment phase is initiated and will be described in Chapter 7.

7

Experiment Design

In this chapter, the experiment phase of the research project is discussed. All the experimentation during this project will be performed using Python-based computer models. During the first phase, the conceptual idea is tested, see Section 7.1. The second experiment phase is described in Section 7.2, and focuses on aircraft separation. During Experiment Phase 3, the full-scale testing setup with multiple aircraft in BlueSky is conducted, this is explained in Section 7.3.

7.1. Experiment Phase I: Small Scale Controller Testing

During the first experiment phase, the bi-level model predictive controller shall be tested and compared to a single level controller using multiple case studies. These experiments will be conducted on a single aircraft, conflict detection and avoidance falls outside the scope of this experiment. The goal of these case studies is to obtain a clear understanding of the possible benefits bi-level controller could have on a single aircraft.

The set-up of the experiment is an aircraft, that navigates through an area filled with random obstacles, towards a certain destination, see Figure 7.1. The aircraft has to stay clear at least 5 nautical miles from all the obstacles. The same simulation iteration, using similar initial conditions and obstacle locations, is run three times. Once for a single level MPC, once for the bi-level controller using one setpoint, and finally for the bi-level controller using two setpoints. The obstacles are added to force the controllers to initiate heading and velocity changes.

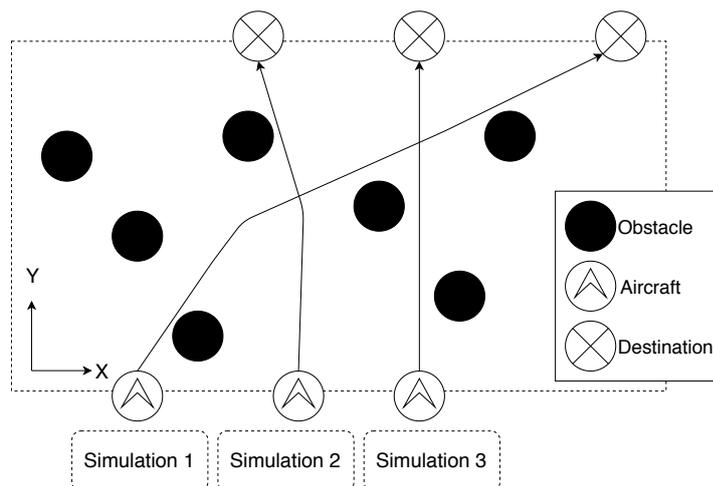


Figure 7.1: Experiment phase I, Aircraft with random initial conditions navigate through an area of random obstacles towards a randomly placed waypoint.

The independent variables are the centralized and decentralized prediction horizons, and the control sample time of each controller. The single level MPC uses the same parameters as the decentralized controller in the

bi-level control architecture, to gain an insight into the influence of the centralized controller. The dependent variables during this experiment are the total fuel consumption, flight time and computation time.

7.2. Experiment Phase II: Conflict Avoidance

The second set of experiments entails the conflict avoidance capabilities of the bi-level MPC. During this experiment, case studies will be conducted with carefully designed initial conditions, meaning that the two aircraft have symmetrical but opposite headings, initial positions, and destinations. The two aircraft will be of the same type and the initial velocity is equal. This will cause a conflict if no resolving action is taken, as can be seen in Figure 7.2.

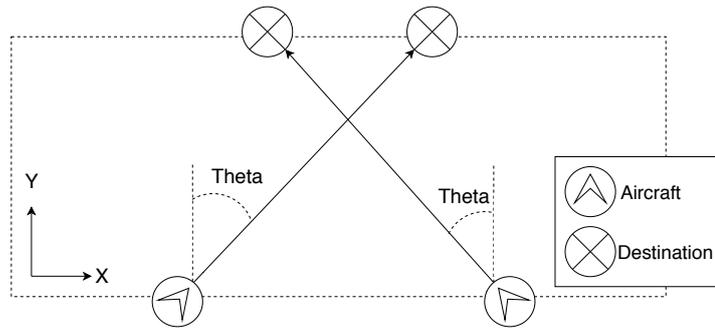


Figure 7.2: Experiment phase II, bi-level separation constraint testing using two aircraft.

For the experiments during the second experiment phase, the same set of independent variables that were used for the first experiment phase will be tested. Additionally, several predefined heading angles are used to evaluate if the collision angle has any effect on the outcome. This angle is increased from 30° until 90° , in 15° steps. Where 90° would result in a head-on collision if no heading adjustments are initiated.

7.3. Experiment Phase III: Multiple Aircraft Simulation

During the last set of experiments, the controller is tested in a more realistic scenario, in an extension of the first experiment. Instead of running simulations with 1 random aircraft, multiple random aircraft have to navigate towards their destination simultaneously. The static obstacles are removed and replaced by dynamic obstacles in the form of other aircraft.

This experiment will be extended by having aircraft start at varying vertical locations, as well as changing horizontal coordinates, as can be seen in Figure 7.3. The starting and finishing positions have to be at least the minimum separation distance apart, such that the initial and final conditions are feasible.

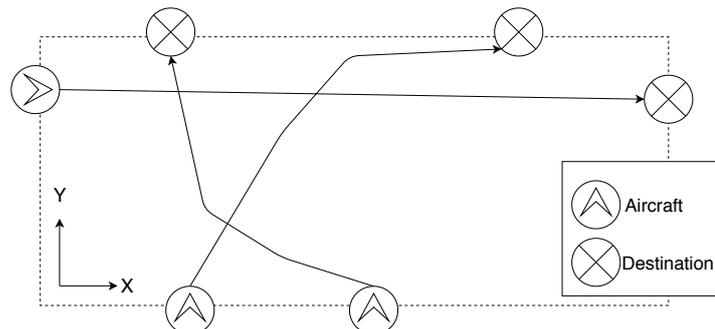


Figure 7.3: Experiment phase III, multi-aircraft testing in BlueSky.

This experiment is conducted to evaluate how the bi-level model predictive controller behaves when controlling higher traffic densities, to provide a conclusion and provide recommendations for future research. The computation time, loss of separation and the ratio between distance flown and the shortest possible distance will be evaluated. Three different traffic densities are to be tested; small (2-4 aircraft), medium (5-10 aircraft) and large (>10 aircraft).

8

Preliminary Experiment Results

In this chapter, two sets of experiments are presented. In Section 8.1, results can be found of a series of static obstacle avoidance tests, when using only a single level decentralized model predictive controller. These tests are conducted to evaluate how the prediction horizon and control sample time affect the obstacle avoidance capabilities of the MPC. The second set of experiments is an extension of the static obstacle avoidance tests, where a single level MPC controller is compared to a bi-level MPC using one or two setpoints constraints for the decentralized controller. The results of the second experiment set can be found in Sections 8.2. In Section 8.3, the handling of a larger constraint area is presented. The effect of the setpoint sizes is discussed in Section 8.2.3. Finally, 3 different methods of implementing the setpoint principle are explained in Section 8.4. In the simulations of both experiments, the model parameters of a Boeing 747, are used in the simulation, these parameters can be found in Table C.1, in Appendix C.

8.1. Experiment 1: Decentralized Static Obstacle Avoidance

During the first experiment set, an aircraft is placed at a starting position [0,0] and is tasked to navigate towards a destination at [100, 100]km. A static obstacle is placed at [50, 50]km. The minimum separation the aircraft needs to maintain with the obstacle is 5NM, creating a circular forbidden area. The cost function in this experiment only includes the absolute distance between the aircraft and the final destination.

8.1.1. Experiment 1: Experiment Setup and Performance Metrics

The decentralized control sample time and the decentralized prediction horizon are varied in each experiment to develop a range of suitable prediction horizons and sample times combinations. The dependent variables are the flight time, fuel consumption, computation time, and feasibility of the solution.

When the aircraft is within a proximity of 5km of its destination the time of flight, the fuel flow and the computation time is determined. Anything after the aircraft reached its destination is not taken into account.

Another metric is introduced to compare the deviation from the shortest path each aircraft has to take. Since the final waypoint is located at [100, 100]km and the starting position is [0, 0]km, the initial heading angle is 45 degrees. The shortest path, in this case, is given by the function:

$$f_{\text{line}}(x) = y \quad (8.1)$$

At every discrete sample time, the Euclidean distance between the aircraft's location and the shortest path line is determined, using:

$$\Delta_{\text{path}} = |y - f_{\text{line}}(x)| \sin \left(90 - \tan^{-1} \left(\frac{x_{\text{final}}}{y_{\text{final}}} \right) \right) \quad (8.2)$$

for the derivation of this formula, see Figure 8.1.

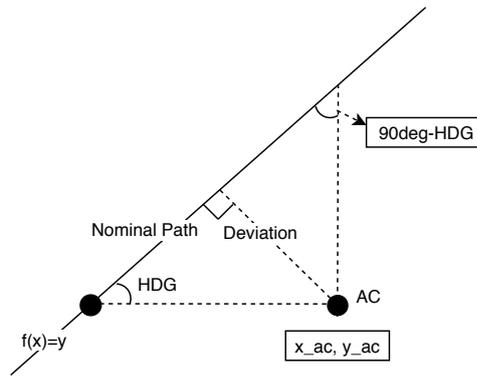


Figure 8.1: Derivation of Equation 8.2.

The initial conditions and constraint boundaries for experiment 1 are given in Table 8.1.

Table 8.1: Initial conditions for experiment 1, decentralized static obstacle avoidance

V_0	250	m/s
x_0, y_0	0, 0	km
φ_0	45	deg
a constraints	-0.5, +0.5	m/s^2
ψ constraints	-25, +25	deg
V constraints	200, 300	m/s

8.1.2. Experiment 1: Simulation Results

The numerical results for each experiment are presented in Table 8.2. The flight time, fuel used, simulation computation time and average path deviation are given for 16 different combinations of control sample times and prediction horizons.

Table 8.2: Results experiment 1, decentralized static obstacle avoidance.

Δt	H_p	Flight Time (s)	Fuel (kg)	Comp. Time (min)	Avg. Deviation (m)
5	5	475	1339	3.61	2930
	10	485	1358	3.76	4692
	15	475	1309	5.17	3542
	20	475	1284	7.33	3472
10	5	490	1354	1.59	4235
	10	480	1299	1.76	3904
	15	480	1245	2.69	3583
	20	480	1184	3.96	3793
15	5	480	1332	0.98	3699
	15	480	1243	1.25	3610
	15	495	1169	1.97	4285
	20	0	0	0	0
20	5	480	1292	0.74	3603
	10	480	1202	1.01	3994
	15	500	1116	1.45	4681
	20	0	0	0	0

The aircraft trajectories corresponding to the first 4 simulations presented in Table 8.2, can be seen in Figures (8.2-8.5).

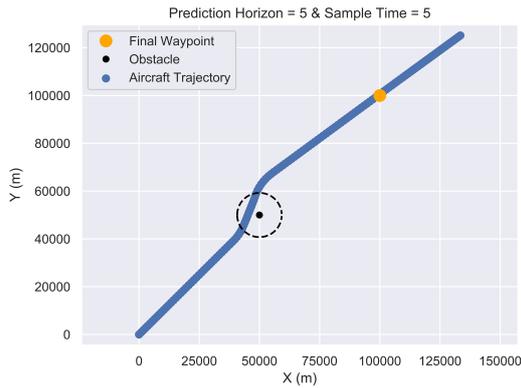


Figure 8.2: Aircraft trajectory with static obstacle, with $H_p = 5$ and control sample time = 5 seconds.

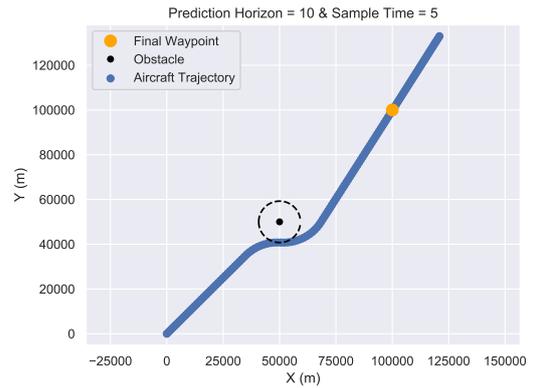


Figure 8.3: Aircraft trajectory with static obstacle, with $H_p = 10$ and control sample time = 5 seconds.

From Figure 8.2, it becomes clear that a look-ahead time of 25 seconds, which is the result of 5 control samples spaced 5 seconds apart, is insufficient for the aircraft to avoid the forbidden area. Every other combination of sample time and prediction horizon resulted in a feasible solution where the minimum separation requirement is achieved.

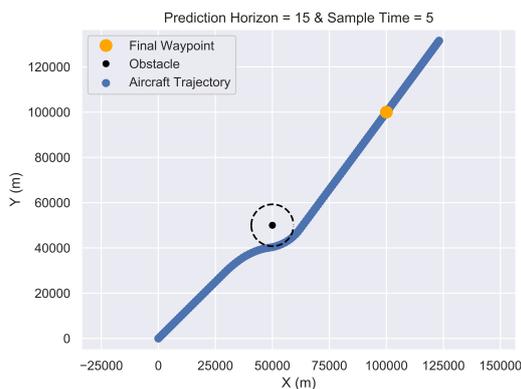


Figure 8.4: Aircraft trajectory with static obstacle, with $H_p = 15$ and control sample time = 5 seconds.

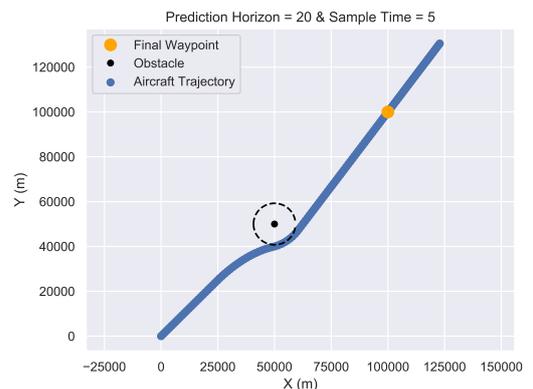


Figure 8.5: Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 5 seconds.

A close-up version of Figure 8.2 and B.4 can be found in Figure 8.6 and Figure 8.7, respectively.

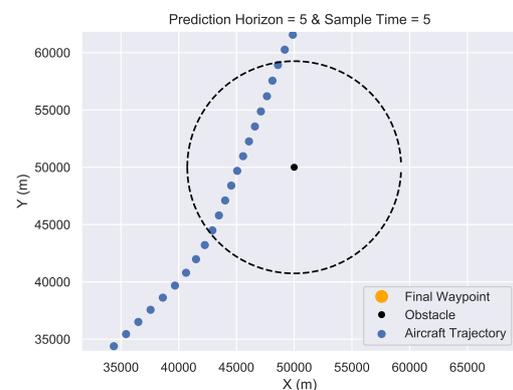


Figure 8.6: Close up of the aircraft trajectory around static obstacle, with $H_p = 5$ and control sample time = 5 seconds.

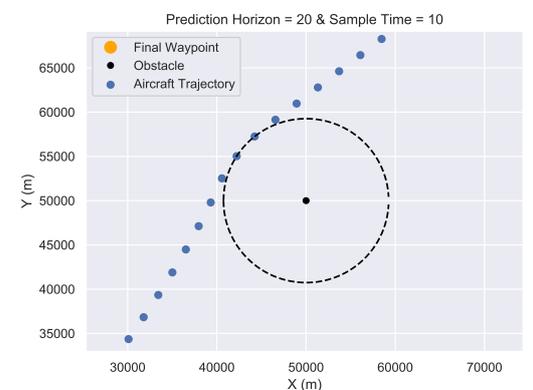


Figure 8.7: Close up of the aircraft trajectory around static obstacle, with $H_p = 20$ and control sample time = 10 seconds.

In Figures (8.8-8.11), four velocity profiles can be seen, corresponding to the first 4 experiments. The first two experiments show a deceleration between 100-300 seconds. This might be explained that a minor constraint violation has occurred that forced the controller to search for a solution.

The velocity profiles verify the working principles of the cost function. The controller initially maximized the velocity to reduce the distance between the aircraft and its destination. However, once the aircraft gets closer to its destination, the controller reduces the aircraft velocity to stay near the destination as long as possible.

When using a larger prediction horizon the deceleration is initiated earlier than for a shorter prediction horizon. This effect can clearly be observed when comparing Figures (8.8- 8.11). The remaining 12 velocity profiles corresponding to the other experiments can be found in Appendix C.

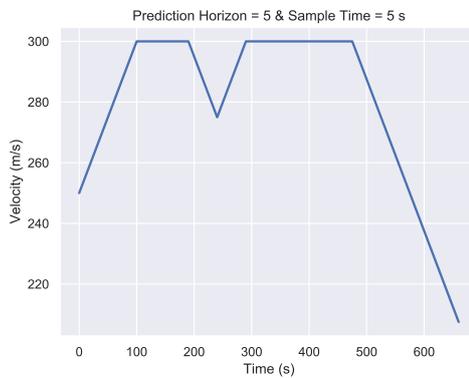


Figure 8.8: Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 5 seconds.

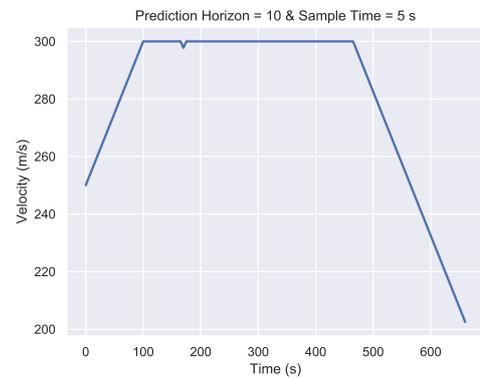


Figure 8.9: Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 5 seconds.

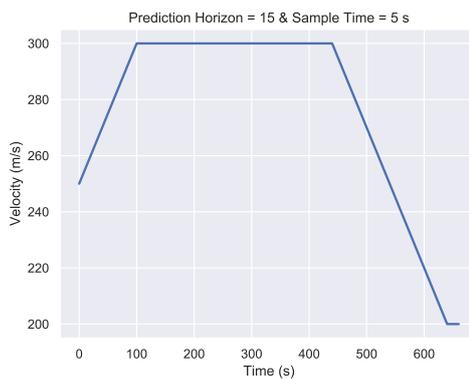


Figure 8.10: Aircraft velocity with static obstacle, with $H_p = 5$ and control sample time = 5 seconds.

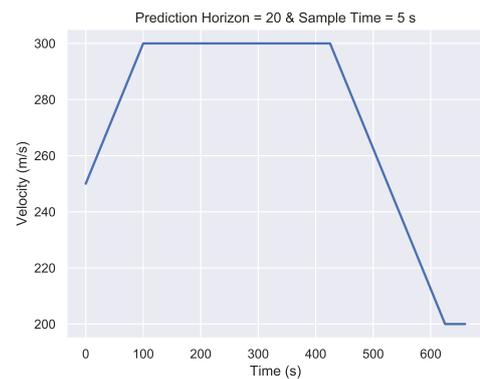


Figure 8.11: Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 5 seconds.

The absolute distance from the initial shortest path, is visualized in Figures (8.12-8.15). The path deviation for all 16 experiments show similar behavior, the deviation from the path should converge towards 9.26km (5NM) around the 5-minute mark of the simulation. All simulations show that the aircraft does not return to the same heading path after the obstacle has been avoided.

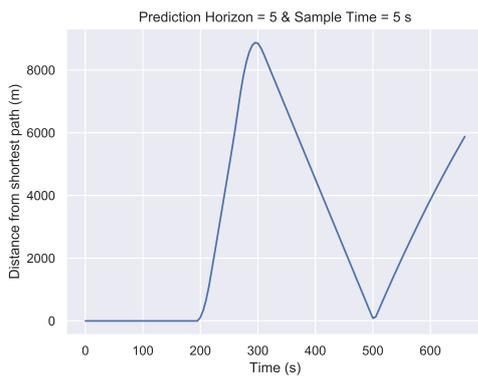


Figure 8.12: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 5$ and control sample time = 5 seconds.

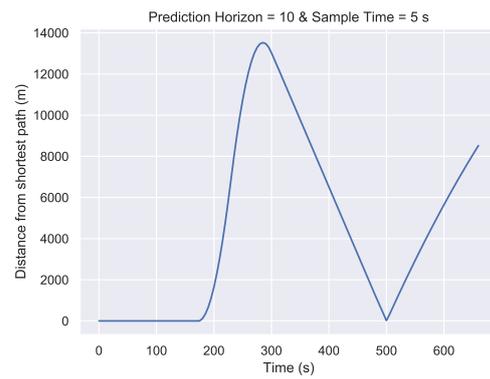


Figure 8.13: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 10$ and control sample time = 5 seconds.

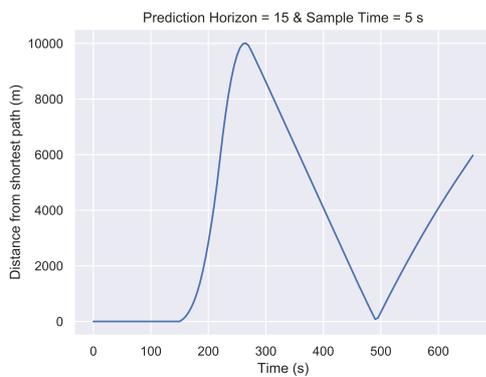


Figure 8.14: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 15$ and control sample time = 5 seconds.

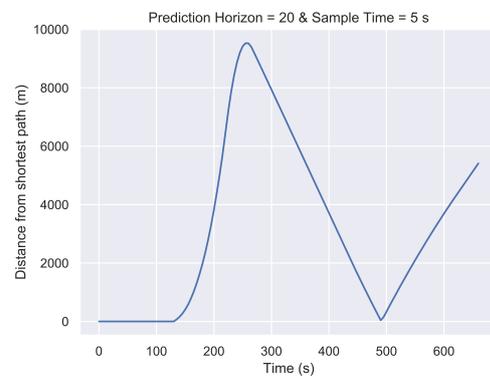


Figure 8.15: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 20$ and control sample time = 5 seconds.

Figures (8.16-8.19), correspond to experiments with a look-ahead time of 300 and 400 seconds. When observing the trajectories of these experiments it can be observed that the aircraft circles around the destination. Due to the relatively large look-ahead time of 300 and 400 seconds, respectively. The controller can predict that when the aircraft flies straight through the waypoint, it will eventually be further away from the destination than when the aircraft starts circling around the final waypoint, with a minimum turn radius.

It can be seen that the final position of the aircraft in these two simulation runs is closer to the destination than the other simulation runs where the aircraft flies in a straight line. The trajectories of the remaining 10 simulation runs can be found in Appendix B.

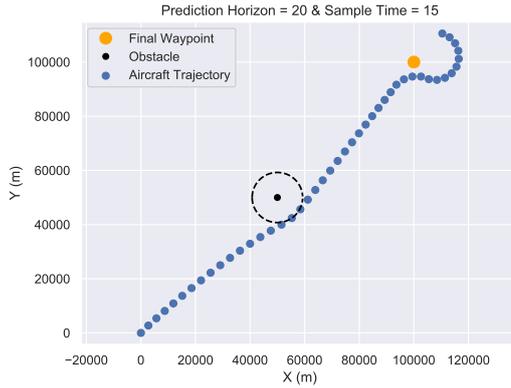


Figure 8.16: Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 15 seconds.

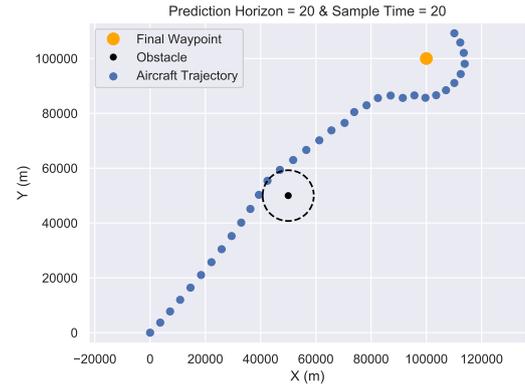


Figure 8.17: Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 20 seconds.

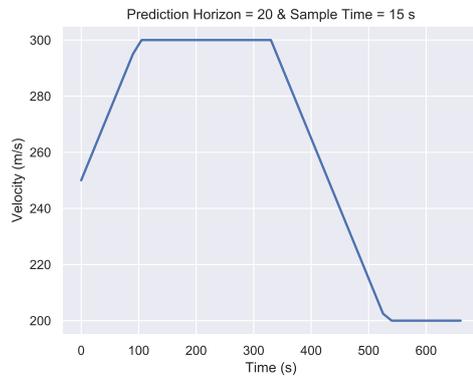


Figure 8.18: Aircraft velocity with static obstacle, with $H_p = 20$ and control sample time = 15 seconds.

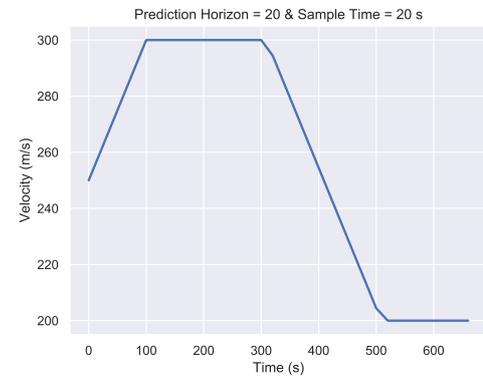


Figure 8.19: Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 20 seconds.

8.2. Experiment 2: Number of Setpoint Determination

In this section, the setpoint integration tests are presented. The difference between using one and two setpoints is explored and simultaneously compared to a decentralized controller using no setpoints, which is effectively a single level decentralized controller. These experiments are performed to determine if a difference in computation time and performance can be detected.

8.2.1. Experiment 2: Setpoint Testing Experiment Setup

In this experiment, an aircraft with known but randomly selected initial condition is placed at a horizontal position at $y=0$ km, which must fly a trajectory to a given position at $y = 200$ km. At the horizontal line $y = 50$ km, 3 coordinates are selected that have to be avoided with a clearance of at least 5NM, creating circular obstacles with a radius of 5NM.

The simulation time is 4 minutes per run such that the final position is far out of reach. This is done to mitigate unwanted behavior where an aircraft circles around its destination, as was observed in Figure 8.16 and Figure 8.19. The initial conditions for the aircraft during these simulations are given in Table 8.3.

Table 8.3: Initial condition for Experiment 2, bi-level obstacle avoidance.

x_0	10 - 90	km
y_0	0	km
V_0	240 - 280	m/s
p_0	-45, +45	deg
x_{final}	10 - 90	km
y_{final}	200	km
a constraints	-0.5, +0.5	m/s ²
ψ constraints	-25, +25	deg
V constraints	$V_0 - 6\%$, $+V_0 + 25\%$	m/s
$N_{\text{obstacles}}$	3	-
$x_{\text{obstacles}}$	15 - 85	km
$y_{\text{obstacles}}$	50	km

Ten scenarios are tested for eight different sets of experiments, the experiment sets can be found in Table 8.4. These eight combinations of prediction horizons and control sample times are carefully chosen, to compare the effect of each parameter. The product of Δt_d and H_{p_d} is equal to two times H_{p_c} , to make sure the second setpoint falls within the prediction horizon of the decentralized controllers. For the simulation using only one setpoint, this results in more freedom at later iterations in the prediction horizon, because there is no setpoint constraint to adhere to.

Table 8.4: Experiment 2, prediction horizon and control sample time settings.

Experiment Number	Δt_c (s)	H_{p_c}	Δt_d (s)	H_{p_d}
Experiment 1	30	5	10	6
Experiment 2	30	5	15	4
Experiment 3	30	10	10	6
Experiment 4	60	5	10	12
Experiment 5	60	5	20	6
Experiment 6	60	5	30	4
Experiment 7	60	10	10	12
Experiment 8	60	10	20	6

In each experiment set, different initial conditions with the different obstacle locations are tested. However, the same set of initial conditions and obstacle locations are repeated for each new experiment set. Similar case studies can, therefore, be compared for different prediction horizon and sample time combinations. Every combination is tested 10 times and the resulting fuel consumption, remaining distance, computation time and the number of constraint violations are recorded.

8.2.2. Experiment 2: Simulation Results

The reason for using obstacles in these tests becomes clear when considering Figures (8.20-8.21). If a straight, obstacle-free, path is available between the initial position and final position, all three controller methods yield a similar path, as can be seen in Figure 8.20. However, when heading changes need to be applied to avoid the forbidden area, the difference between each control method becomes evident, examples can be found in Figures (8.21-8.23).

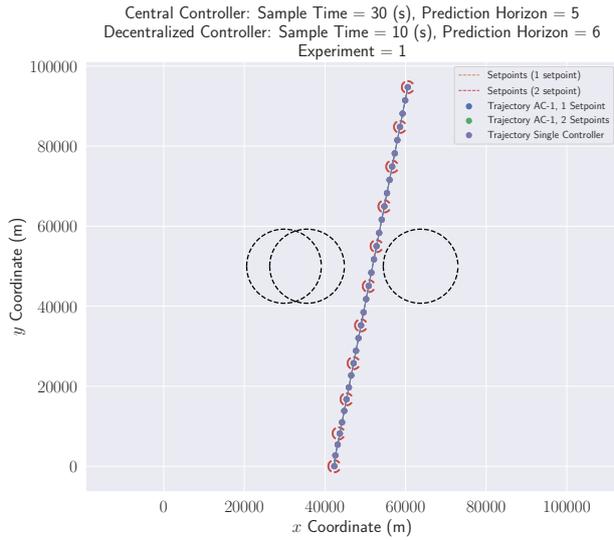


Figure 8.20: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 2.

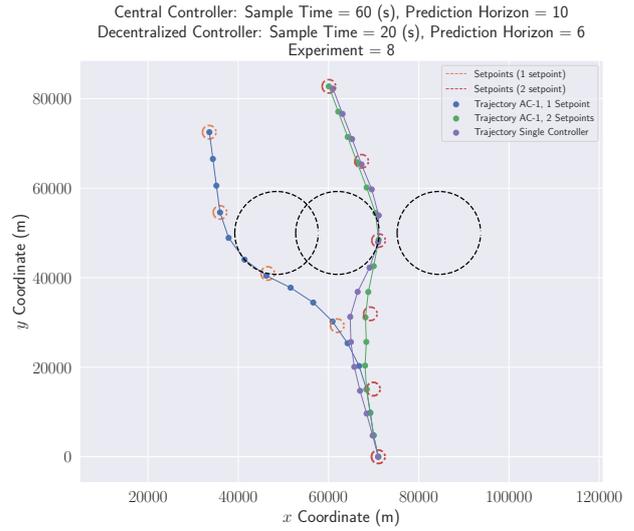


Figure 8.21: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 8, initial condition iteration 1.

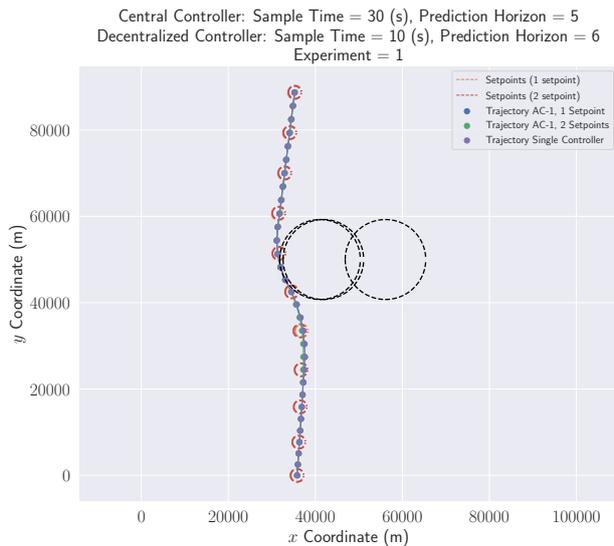


Figure 8.22: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 6.

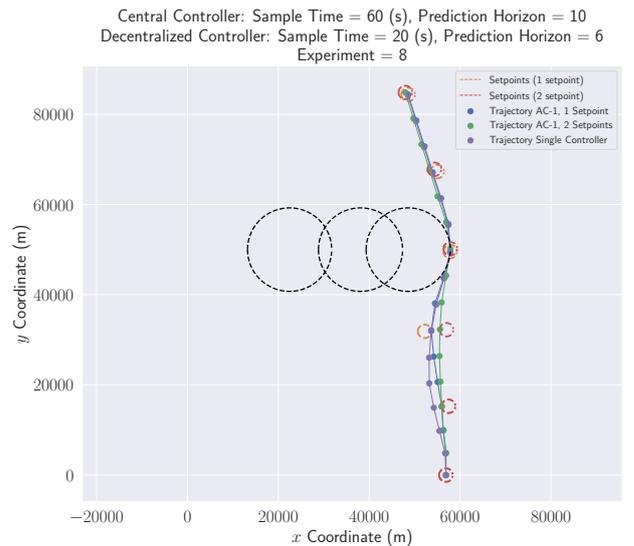


Figure 8.23: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 8, initial condition iteration 10.

Different experiment sets for the same initial conditions can show different chosen trajectories. This becomes apparent when comparing Figure 8.24 with Figure 8.25. Figure 8.24, shows the simulation result for experiment set 1 and Figure 8.25 shows the exact same initial conditions and obstacle placement using the parameters from experiment 7.

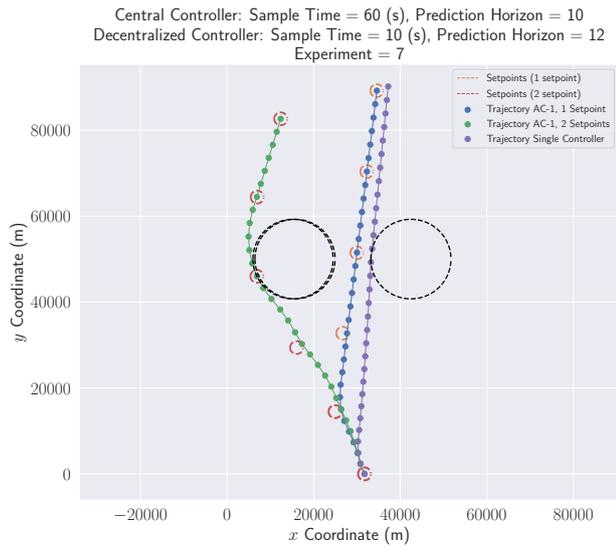
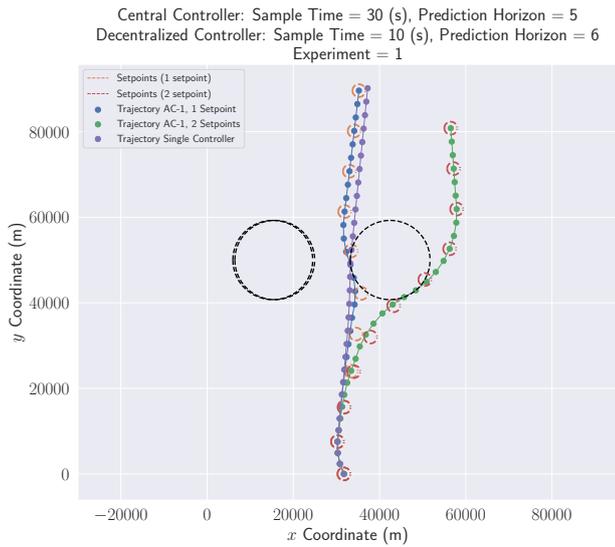


Figure 8.24: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 5.

Figure 8.25: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 7, initial condition iteration 5.

Assessing if the forbidden area constraints have been violated is one of the key performance metrics during these tests. Two examples of flights where at least one of the control methods violate the forbidden area constraint are shown in Figures (8.26-8.27). In Figure 8.26, only the method using 2 setpoints was able to avoid the obstacles, while in the iteration shown in Figure 8.27, the single level controller, as well as the two-setpoints controller, did not violate the constraint.

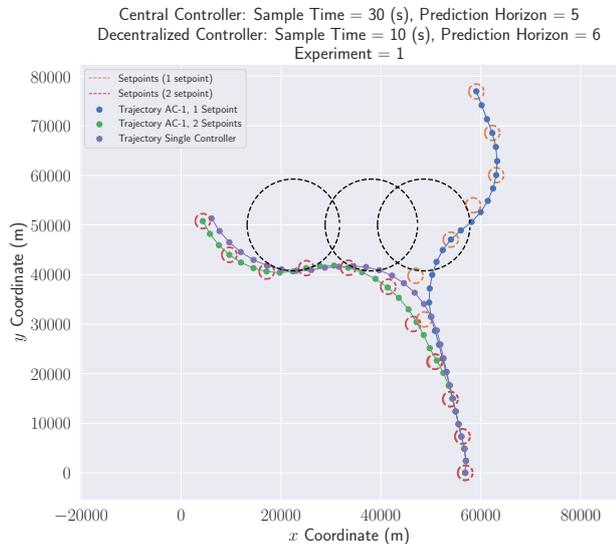
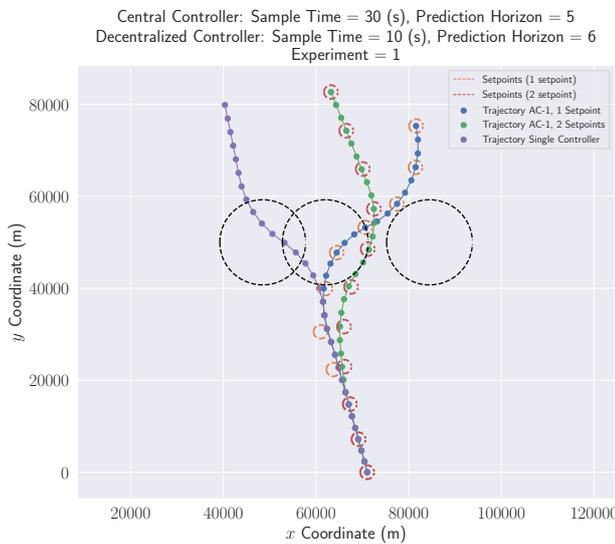


Figure 8.26: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 1.

Figure 8.27: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 10.

In Figure 8.28, the number of runs in which the forbidden area was not successfully avoided is reported. From the violation count, it can be concluded that the first 3 experiment sets all violate the forbidden area constraints. Experiment number 3 has the same look-ahead time window as experiments 4, 5, and 6 of five minutes. The experiment runs that use a control sample time of 60 seconds at the centralized level (4, 5, 6, 7, and 8) outperform the 30 seconds control sample time runs (1, 2 and 3) in terms of constraint adherence.

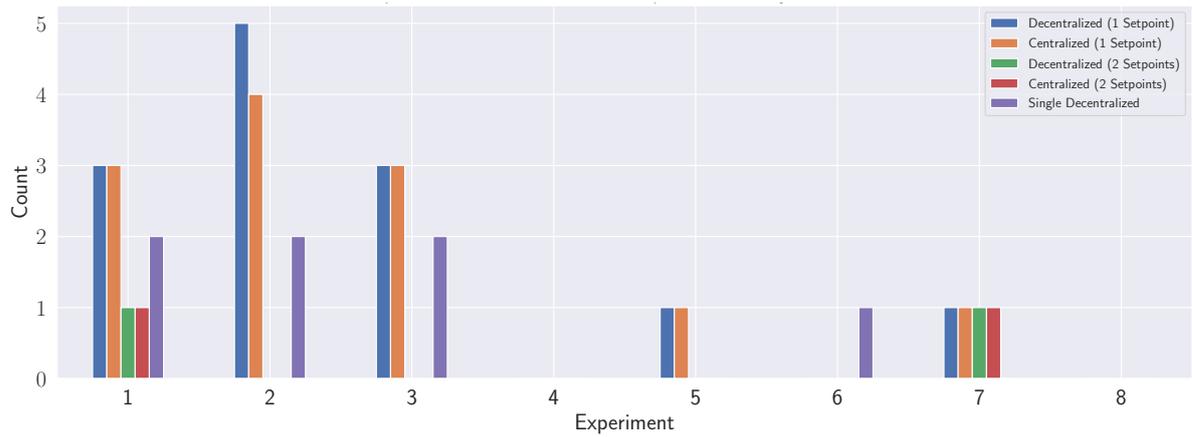


Figure 8.28: Forbidden area violation count with a setpoint boundary of 1500m.

Two metrics that were evaluated during this experiment are the fuel consumption and the remaining distance to the final waypoint, shown in Figure 8.29 and Figure 8.30, respectively. In terms of fuel consumption, each experiment set yields comparable values. This result can be explained because the simulation is ended after four minutes, far before the final destination has been reached. Every aircraft will, therefore, be cruising at the maximum velocity, yielding a similar fuel usage.

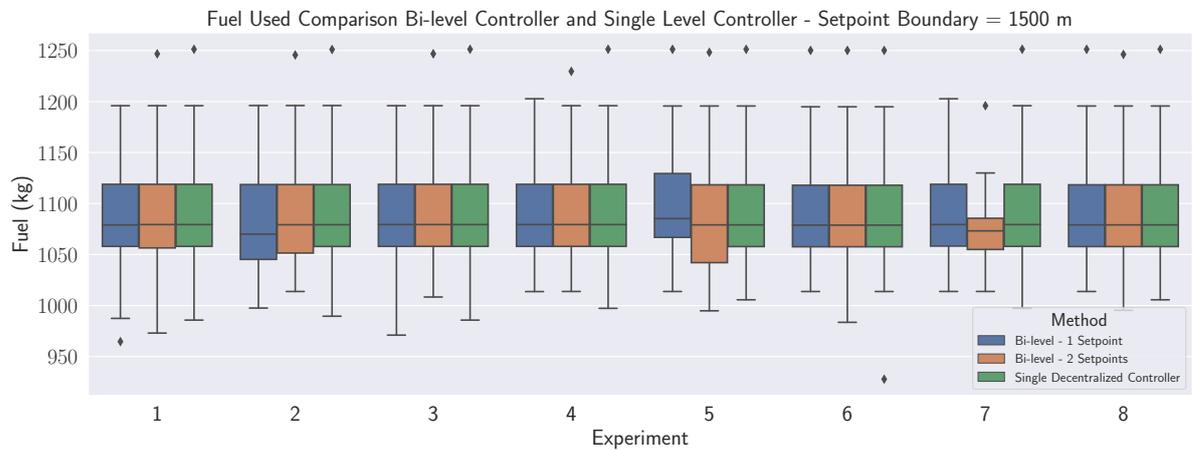


Figure 8.29: Fuel consumption per simulation iteration for 8 experiment sets.

The remaining distance spread in Figure 8.30, seems to be higher for the experiment that uses two setpoints for the decentralized level.

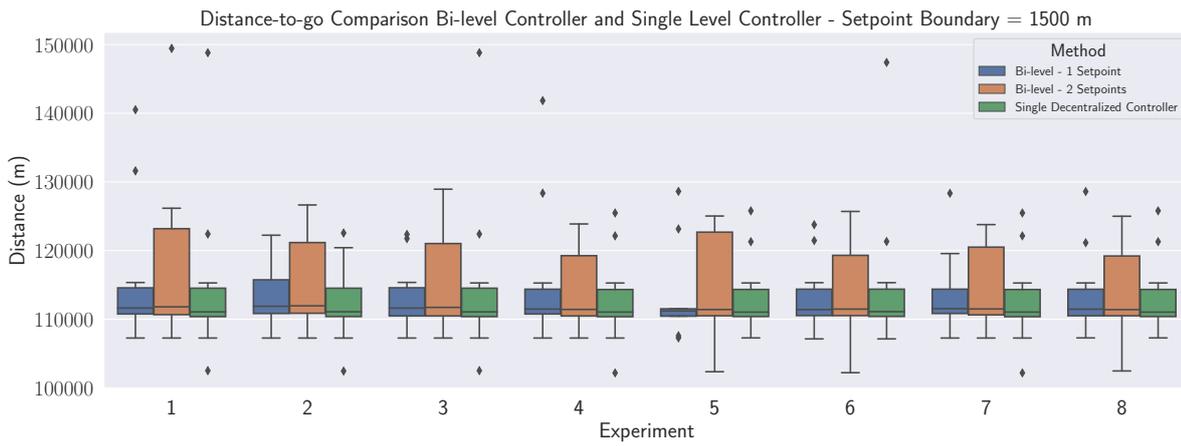


Figure 8.30: Remaining distance to final destination per simulation iteration for 8 experiment sets.

The difference originates from the different trajectories the controllers pick for zero, one or two setpoints, a few examples are presented in Figures (8.31-8.34).

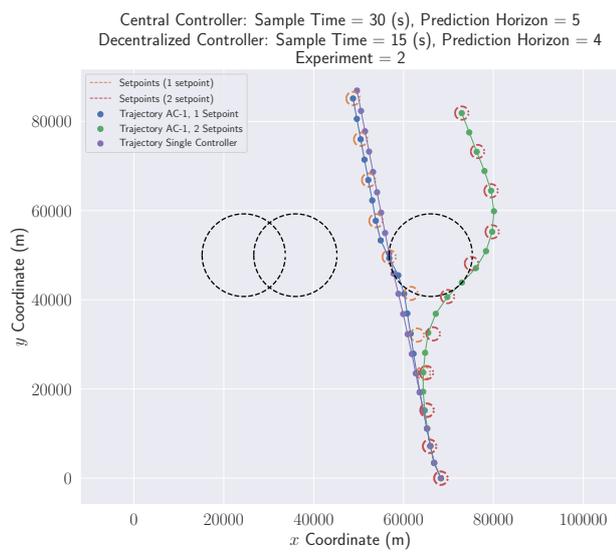
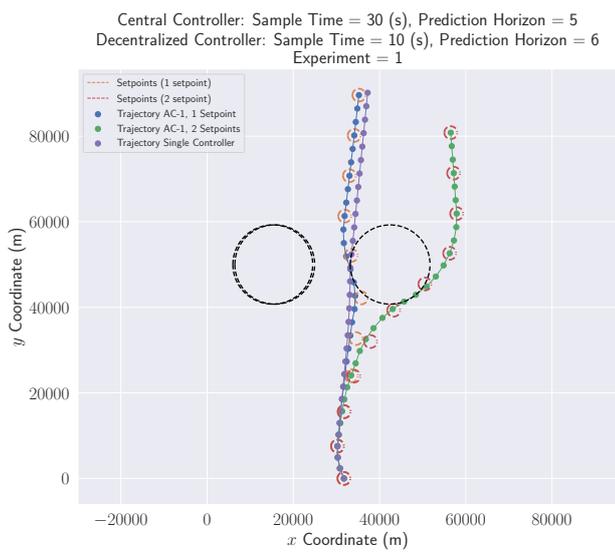


Figure 8.31: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 1, initial condition iteration 5.

Figure 8.32: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 2, initial condition iteration 4.

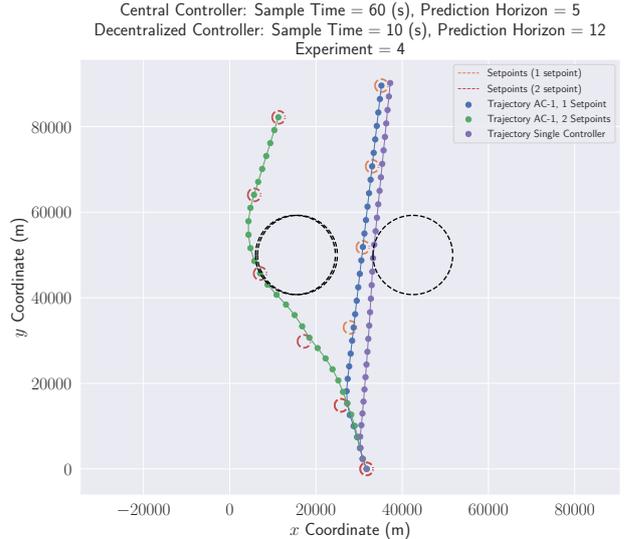
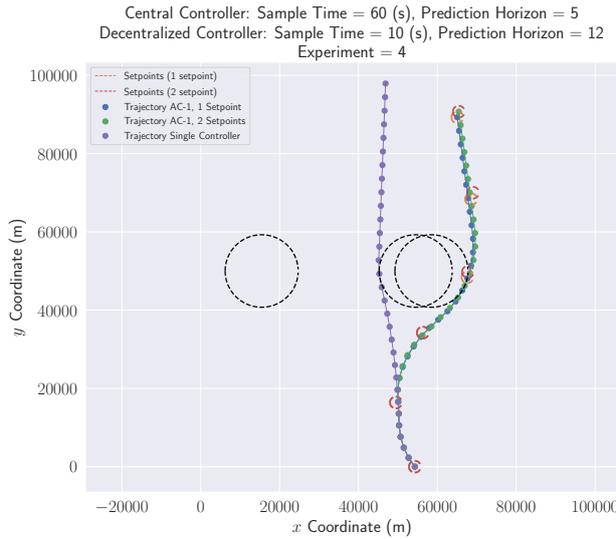


Figure 8.33: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 4, initial condition iteration 8.

Figure 8.34: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 4, initial condition iteration 5.

Figure 8.35, shows the computation times for each experiment set. It can be concluded, that the computation times for experiments 3, 4, 7, and 8 are more than twice as long as for the other experiment sets. Especially, during experiment 7, the computational effort is relatively high. This can largely be contributed to the longer prediction horizons that were used during these simulation runs. However, there is no clear evidence that this experiment sets also outperforms the faster sets in terms of controller performance when considering Figure 8.29 and Figure 8.30.

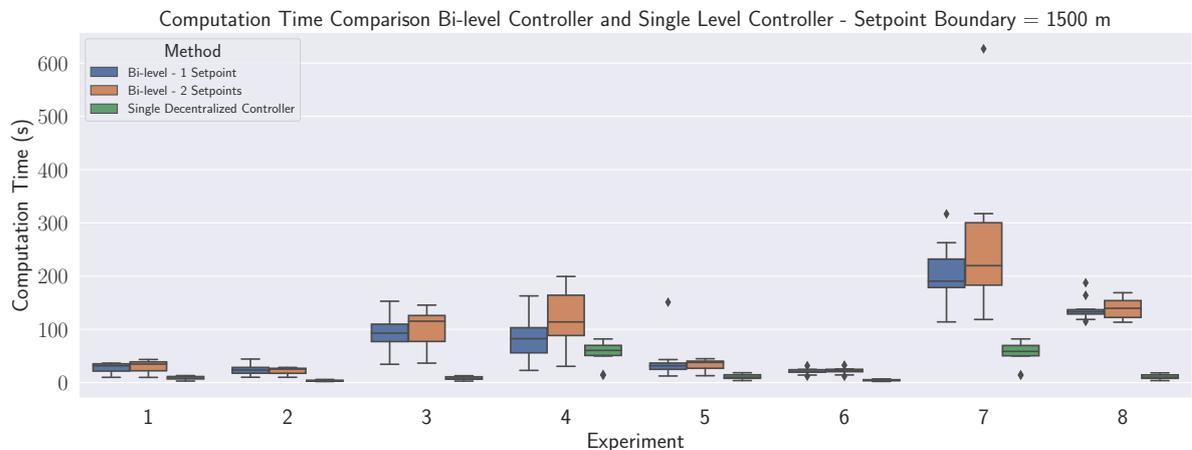


Figure 8.35: Computation times per simulation iteration for 8 experiment sets.

8.2.3. Experiment 2: Setpoints Boundary Size Influence

In this section, the effect of changing the boundary sizes of the maximum deviation from the setpoint is tested. The same ten iterations for the 8 experiment sets, presented in Section 8.2 are run for a setpoint deviation of 1000 meters and 2000 meters (instead of the already tested 1500m, in the previous section. To compare the relative performance of these three setpoint deviation values, the computation time and the forbidden area setpoint violation are compared.

When comparing the computation time results presented in Figure 8.35, with Figures (8.36-8.37) no clear difference is found, the mean for all three boundary sizes test are almost similar. Only during the 7th experiment set for boundary size 1500m a large outlier of ten minutes is detected.

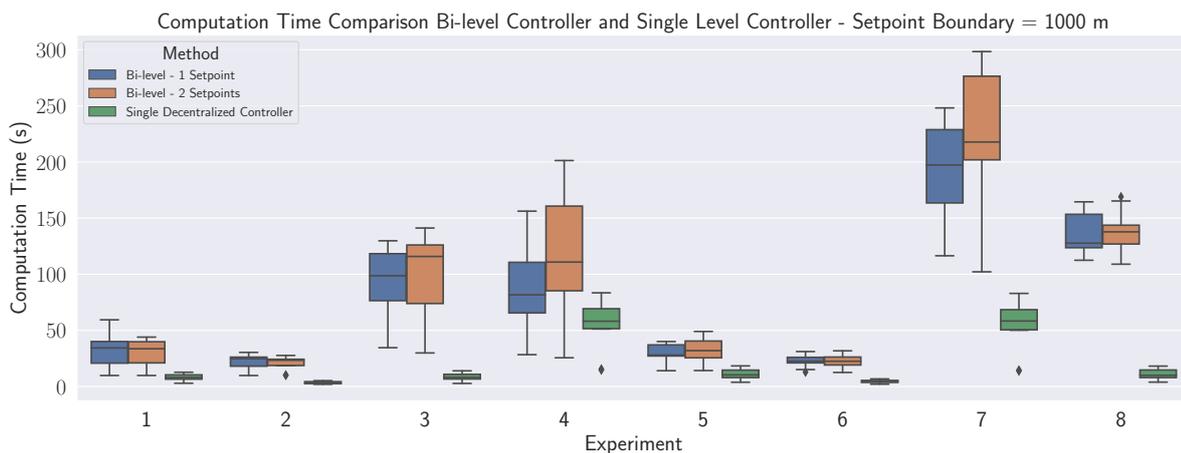


Figure 8.36: Computation times for a maximum setpoint boundary size of 1000 meters.

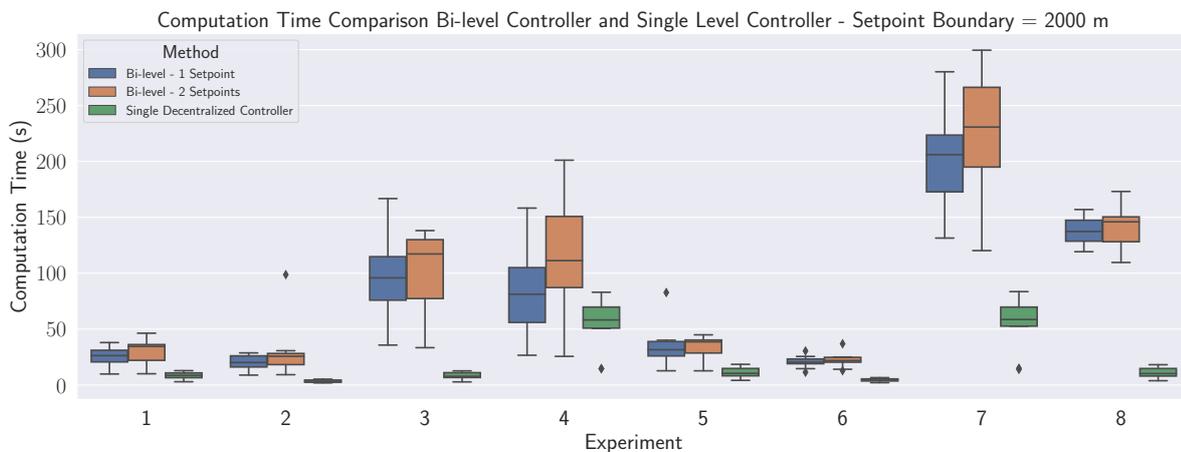


Figure 8.37: Computation times for a maximum setpoint boundary size of 2000 meters.

The feasible results for both tests are affected by changing the setpoint size. The experiment using the boundary size of 1000 meters fails to find a feasible solution 4 out of 10 runs for the first experiment, see Figure 8.38. While the 2000 meters boundary test only fails 2 out of 10 runs for the first experiment set. Furthermore, it can be seen in Figure 8.39, that every experiment violates the forbidden area constraint at least once, whereas this was not the case for the 1000 meters tests.

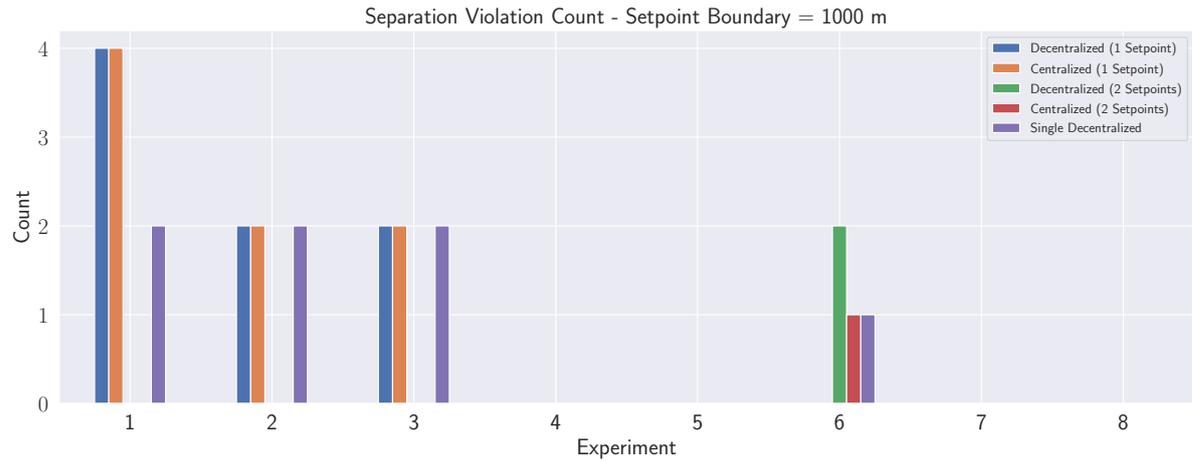


Figure 8.38: Forbidden area violation count with a setpoint boundary of 1000m.

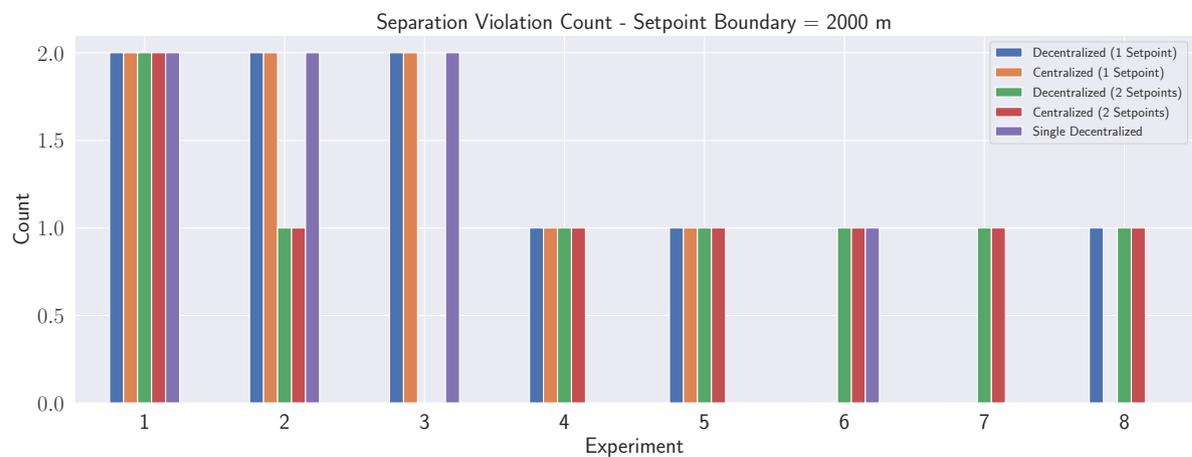


Figure 8.39: Forbidden area violation count with a setpoint boundary of 2000m.

8.2.4. Experiment 2: Increase in Experiment Iterations

In this Section, the results from a similar experiment with a larger iteration count are presented. The experiment sets 2, 5 and 6 are repeated 100 times to present a conclusion based on more data. Experiments 5 and 6 were selected because of the promising results in forbidden area avoidance. Experiment 2 was selected because it showed a large difference between the different controller methods. These experiments yielded the lowest computation times and were, therefore, best suited for an increase in simulation iterations. The experiment parameters are repeated in Table 8.5.

Table 8.5: Experiment sets for 100 iterations.

Experiment Number	Δt_c (s)	H_{p_c}	Δt_d (s)	H_{p_d}
Experiment 1	30	5	15	4
Experiment 2	60	5	20	6
Experiment 3	60	5	30	4

A violin plot showing the computation times for the three different control methods can be found in Figure 8.40. From this plot can be concluded that the spread in computation times for one and two setpoint simulations is relatively similar. The computation time for a single level controller is shorter, a logical result considering that there is no centralized optimization taking place.

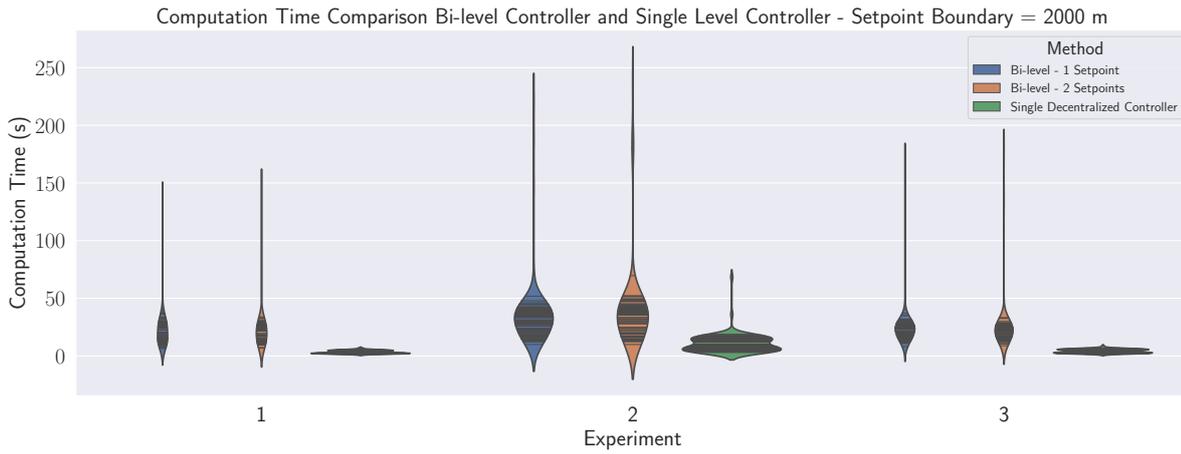


Figure 8.40: Computation time for three different control methods using 100 simulation iterations.

Several conclusions can be drawn from the area violation performance, for the higher number of simulation runs, presented in Figure 8.41. During experiment set 1, only one centralized setpoint was placed within the forbidden location and two instances of the decentralized controller violated the constrained. In contrast, more than 10% of the single setpoint runs yielded infeasible solutions.

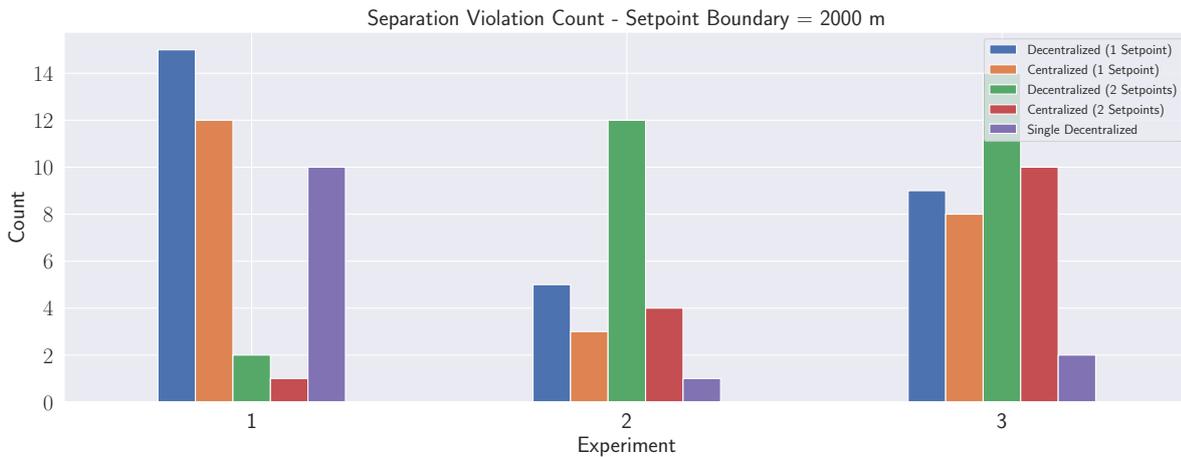


Figure 8.41: Area constraint violation count using 100 simulation iterations.

For experiment set 2 and 3, the opposite result is visible, and the one-setpoint runs seem to outperform the two setpoint-methods in terms of feasibility. This is partly a result of the longer sample time of the centralized controller.

This longer sample time places the setpoints far enough apart such that a forbidden area can fall in between two setpoints, this forces the decentralized controller to steer through the forbidden area, as can be seen in Figure 8.42. Since the decentralized controller must adhere to conflicting setpoint constraints and the forbidden area constraint, the infeasible unwanted situation can occur as can be seen in Figure 8.43. More research must be conducted regarding these results to draw a conclusion.

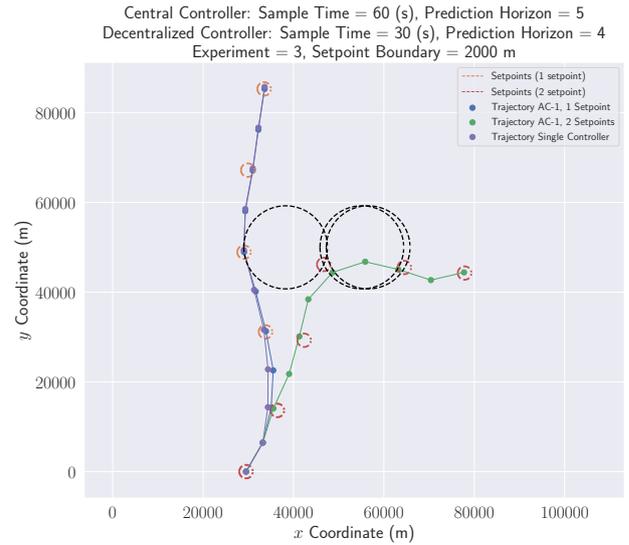
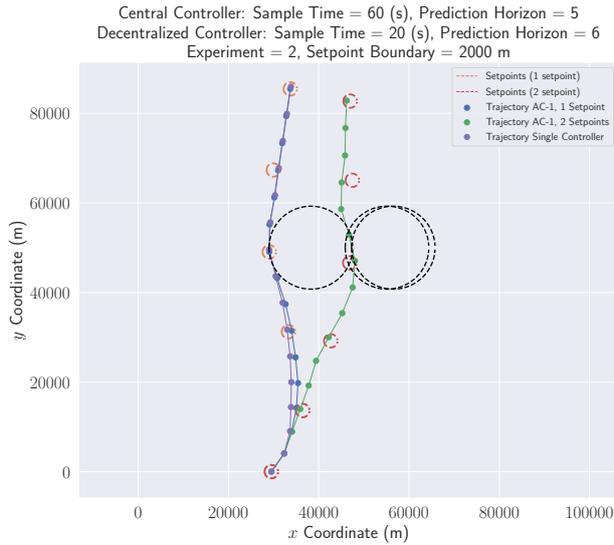


Figure 8.42: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 2, initial condition iteration 16.

Figure 8.43: Three aircraft trajectories for 0, 1 or 2 setpoints, experiment set 3, initial condition iteration 16.

8.3. Experiment 3: Larger Forbidden Area

To test if the size of the centralized control sample time is indeed an issue, several tests were conducted to see the capability of handling larger forbidden areas by the controllers. Figure 8.44 and Figure 8.45, show 2 example trajectories of this experiment.

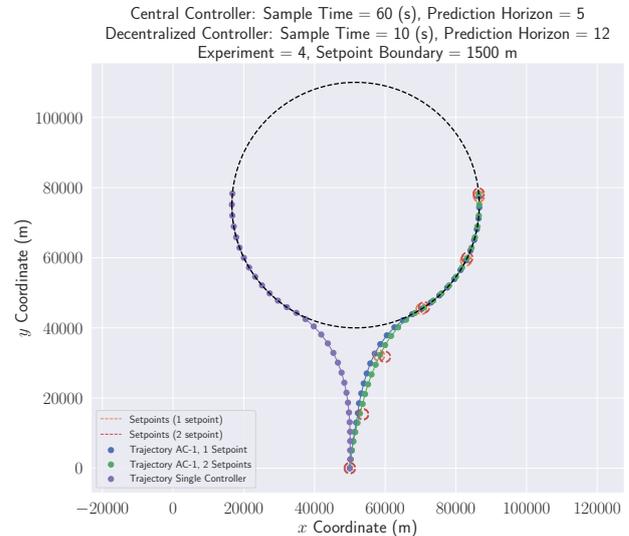
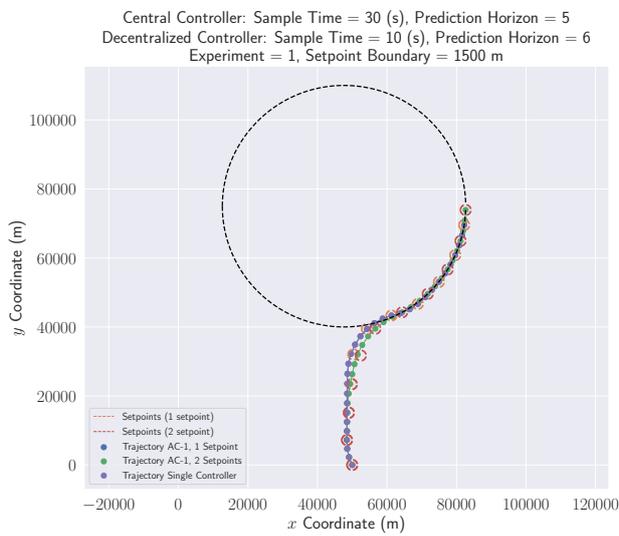


Figure 8.44: Larger forbidden area, experiment set 1, initial condition iteration 3.

Figure 8.45: Larger forbidden area, experiment set 4, initial condition iteration 8.

From the constraint violation count, presented in Figure 8.46, can be concluded that the larger forbidden area causes fewer feasibility issues as a result of a centralized sample time being too large.

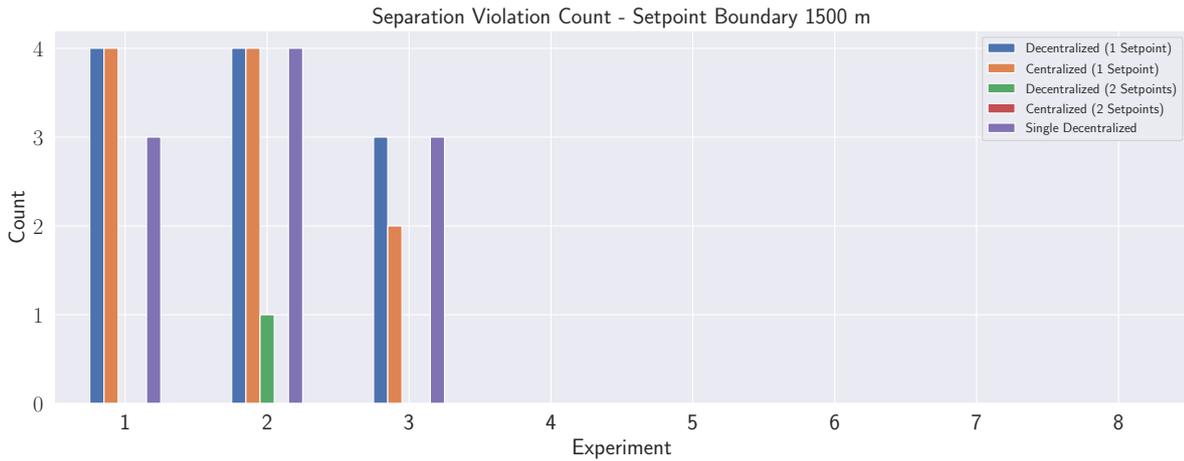


Figure 8.46: Separation violation count for large forbidden area experiment.

8.4. Experiment 4: Single Setpoint Prediction Window Influence

The larger forbidden area experiment presented in Section 8.3 is expanded upon during the tests presented in this section. In the previous experiments (Sections 8.2.1-8.2.3), the decentralized controller used the same controller sample time and prediction horizon, for both the one- and two-setpoint methods. Using the same parameters did entail that the controller using only one setpoint constraint had no setpoint restriction for the second part of the optimization sequence.

To clarify with an example, when using $\Delta t_c = 60$ s the second setpoint corresponds to 120s. When $\Delta t_d = 30$, H_{pd} has to be 4, such that the setpoint falls within the time window of the decentralized controller. If only one setpoint is used, $H_{pd} = 2$ would suffice. Now that $H_{pd} = 4$ is used, the third and fourth instance of the optimization corresponding to 90 en 120 seconds do not have this extra setpoint constraint and have more freedom, which usually results in a heading change towards the final destination, which might be a different direction than the second setpoint.

It is expected that this extra freedom might have resulted in a worse solution when using only one setpoint. Since the decentralized controller has less information about the general direction of the future centralized setpoints.

8.4.1. Experiment 4: Final Experiment Setup

This final experiment setup was conducted. During these tests, one circular obstacle is used with a radius of 40km. The simulation length is increased to 6 minutes and three sets of controller parameters are simulated 25 times.

Table 8.6: Experiment sets for testing the influence of setpoint method options 1, 2 and 3.

Experiment Number	Δt_c (s)	H_{pc}	Δt_d (s)	H_{pd} - Option 1 & 2	H_{pd} - Option 3
Experiment 1	30	5	15	4	2
Experiment 2	60	5	15	8	4
Experiment 3	60	5	20	6	3

The experiments sets presented in Table 8.6 are repeated three times to compare every possible combination of applying the setpoint constraints. The three different decentralized controller options are visualized using Figure 8.47.

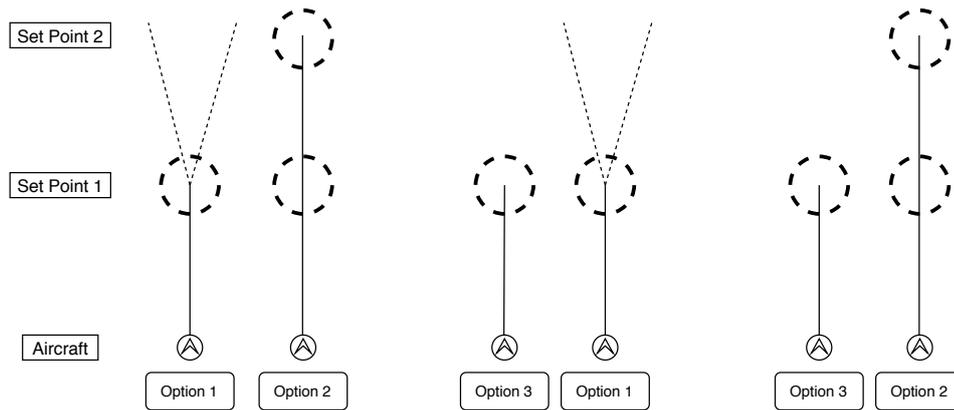


Figure 8.47: Representation of comparison tests of three decentralized controller setpoint methods.

Option 1 and option 2 are a representation of the one-setpoint and two-setpoints methods used, in Sections (8.2.1-8.2.3). This means that the look-ahead window of option 1 was twice as long as the time necessary to reach the first setpoint.

Option 3, uses a one-setpoint decentralized controller with half of the prediction horizon used in the previous two methods. Option 3 is compared to both previously used methods, options 1 and 2. The centralized controller is the same for each of the experiments.

8.4.2. Experiment 4: Final Experiment Setup

The constraint violation results are presented in Figure 8.48. For comparison purposes, the same simulation instance is also performed with a single level controller using either the longer prediction horizon of option 1 or the shorter prediction horizon of option 3.

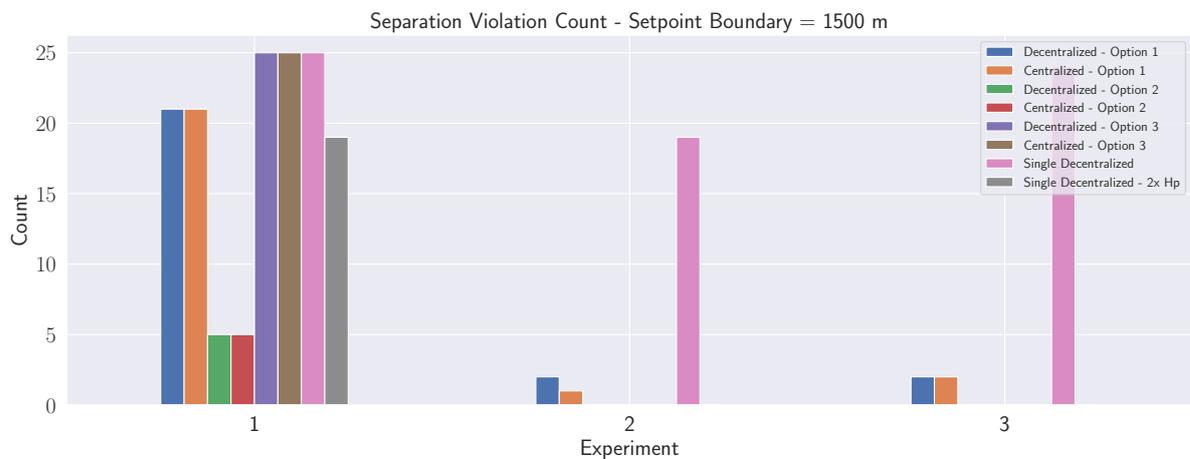


Figure 8.48: Constraint violations for the bi-level controller option 1, 2 and 3 and two versions of a single level controller.

From Figure 8.48, can be concluded that option 1 and option 3, both performed worse than the two-setpoint method (option 2). Furthermore, it can be seen that the single level controllers have a comparable obstacle avoidance capability to options 1 and 2. From this result can be seen that the decentralized prediction horizon is too small to avoid the obstacle and that using just one setpoint is not adequate to improve the performance. Experiment one indicates the benefit of using two setpoints when comparing the results to the single-level controllers.

When considering experiments 2 and 3 it can be observed that option 2 and option 3 did not cause a single violation of the forbidden area constraint, in any of the 25 runs. Whereas option 1 had violated the constraint twice for each experiment.

It is also interesting to see that option 3 performed slightly better than option 1, during experiment 2 and 3, even though the prediction horizon is smaller.

It should be noted that the single decentralized controller, using the same prediction horizon as controller option 3, performed worse by trespassing the forbidden area 19 times. Whereas option 3, succeeded in avoiding the obstacle for each simulation. This is remarkable since the single decentralized controller and the one-setpoint bi-level controller use the prediction horizon and control sample time on the decentralized level, here the influence of the centralized controller becomes evident.

From these three experiments can be concluded that having a prediction horizon that extends further than the setpoint first setpoint constraint is only beneficial when the second setpoint is implemented as an additional constraint. The larger prediction horizon using one setpoint (option 1) even performs worse than the shorter prediction horizon used by option 3, in some cases. Furthermore, the evidence is clear that the additional information provided by the centralized controller reduces the violation of the forbidden area constraint tremendously, compared to a single level decentralized model predictive controller.

Finally, it can be observed that the single decentralized controller violated the constraint 25 out of 25 times during experiment 3. In contrast to only 19 out of 25 times during experiment 2. This is remarkable since they both have a look-ahead time of 60 seconds, yet it appears that the courser time step of experiment set 3, yielded worse results. This proves the concept methodology that adding more detail on the decentral level might increase the performance of the combined system.

9

Conclusion

In this preliminary report, a bi-level controller was introduced with an integrated centralized model predictive controller and a decentralized model predictive controller. The bi-level controller was applied for air traffic control applications and aims to reduce fuel use and flight time while adhering to all safety constraints.

This research aimed to investigate the integrating link between two model predictive control levels that operate with different prediction horizons using different control sample times, yielding varying levels of detail. The benefit of combining a centralized and decentralized control level to a single decentralized or centralized controller was explored. The research question presented in this report read,

“Can a synergy be found between a fast-rate and slow-rate model predictive controller for air traffic control applications, where the benefits of each controller are combined to improve the safety, fuel consumption and flight time compared to a single level model predictive controller?”

This question is partly answered by the results found during the development and the preliminary testing phase. Synergy is achieved between two operational MPC control levels. Both controllers have been integrated and are capable of cooperatively guiding an aircraft past forbidden areas towards their destinations. The developed control system has been tested and was compared to a single decentralized model predictive controller. The results were promising and the bi-level controller performs better than the single-level controller in terms of constraint adherence. In future research, the relative performance based on flight time and fuel usage will be investigated further.

The safety constraint that has already extensively been tested during the preliminary testing phase focuses on flying around forbidden areas. Several controller prediction horizon and control sample time combinations were tested, regarding this constraint. Besides, three separate options of combining the prediction horizon length with either one or two setpoints were investigated.

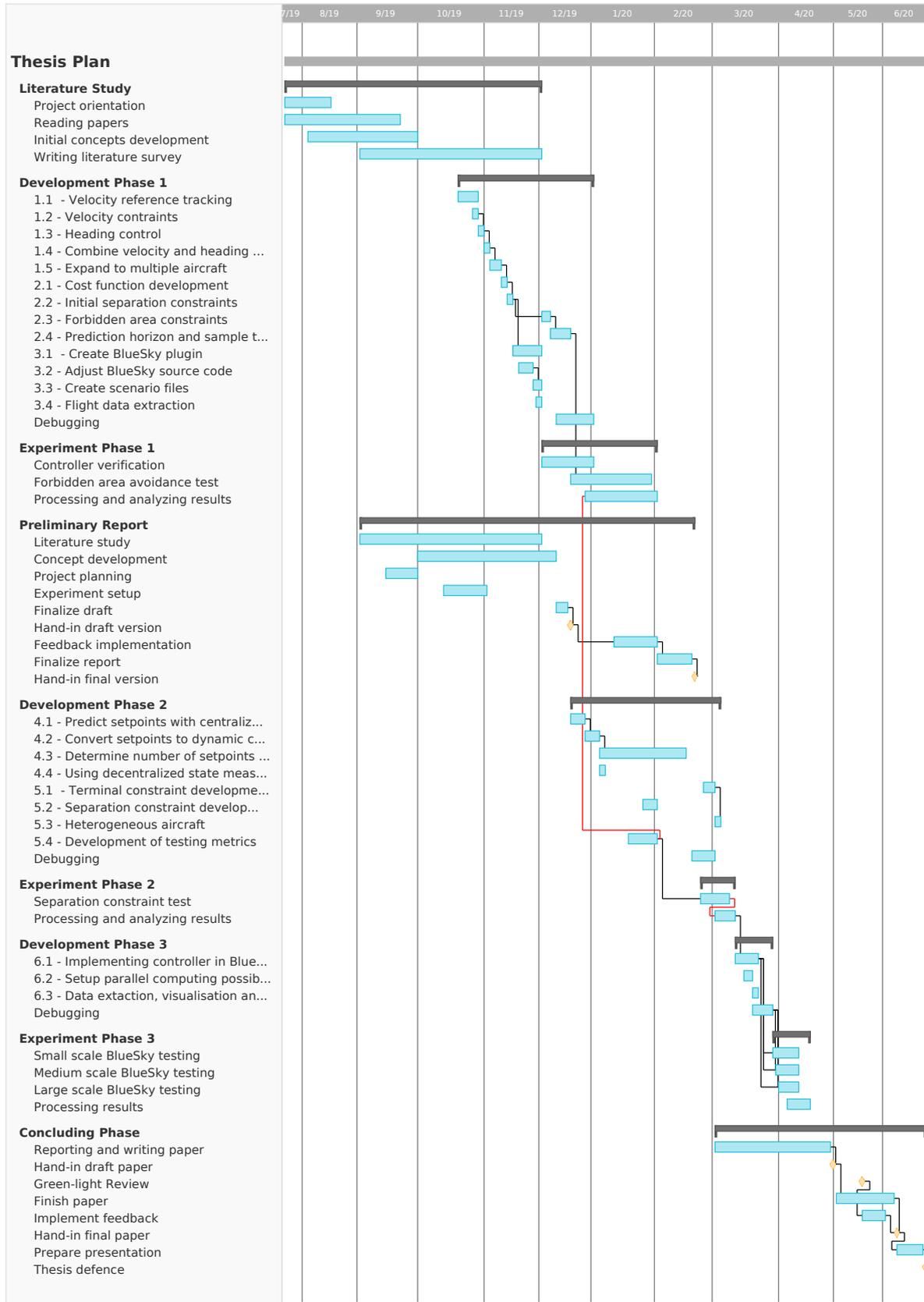
From these tests became clear that using two setpoints is beneficial over the use of one setpoint and that the prediction horizon should not exceed the first setpoint when only using one setpoint. Furthermore, more research must be conducted to determine how the flight time and fuel usage is effected before a trade-off can be made.

The projected result of this Thesis will include a clear answer to all aspects of both research questions. This means that the metric for comparing fuel flow and flight time will become increasingly important. Before these parameters become truly meaningful, terminal constraints need to be developed.

Furthermore, experiment phases II and III will be executed to test the performance when multiple aircraft need to be controlled and separated. During these testing phases, disturbance (wind) handling will be explored.

A

Appendix A: Gantt Chart



B

Appendix B: Initial Simulation Results

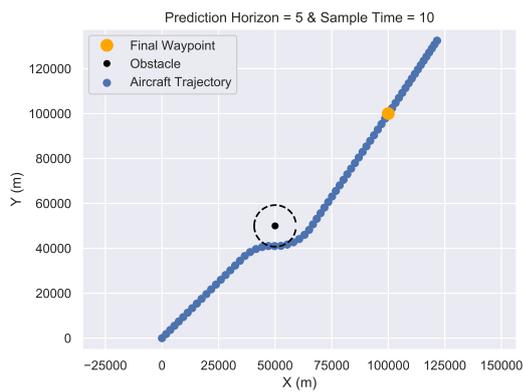


Figure B.1: Aircraft trajectory with static obstacle, with $H_p = 5$ and control sample time = 10 seconds.

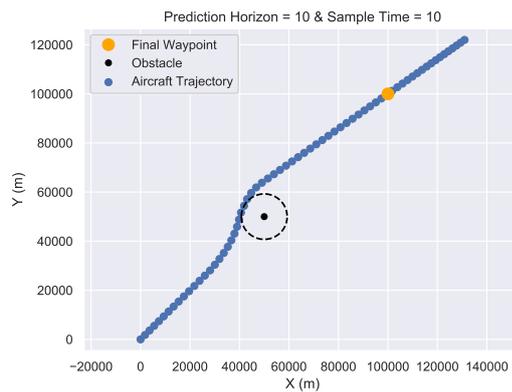


Figure B.2: Aircraft trajectory with static obstacle, with $H_p = 10$ and control sample time = 10 seconds.

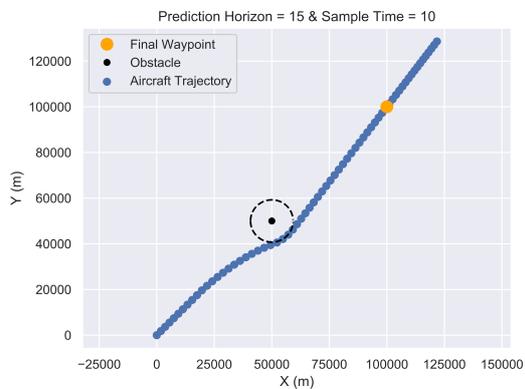


Figure B.3: Aircraft trajectory with static obstacle, with $H_p = 15$ and control sample time = 10 seconds.

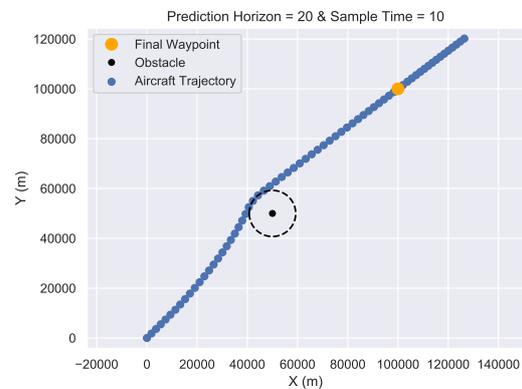


Figure B.4: Aircraft trajectory with static obstacle, with $H_p = 20$ and control sample time = 10 seconds.

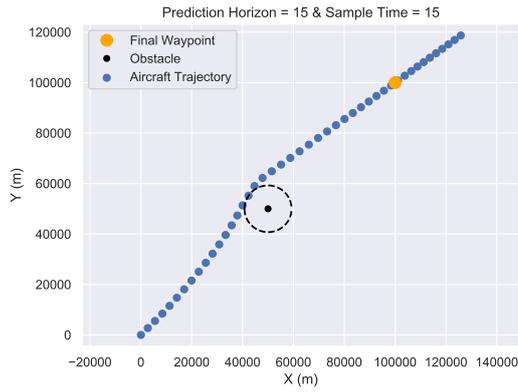


Figure B.5: Aircraft trajectory with static obstacle, with $H_p = 15$ and control sample time = 15 seconds.

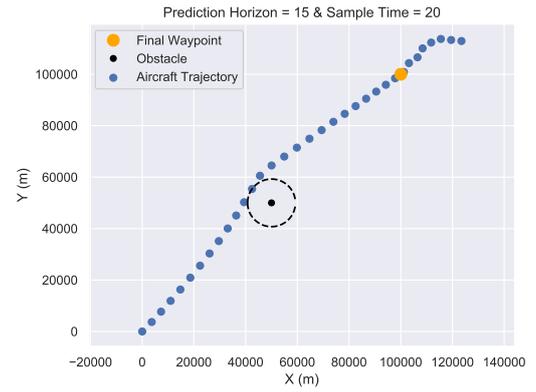


Figure B.6: Aircraft trajectory with static obstacle, with $H_p = 15$ and control sample time = 20 seconds.

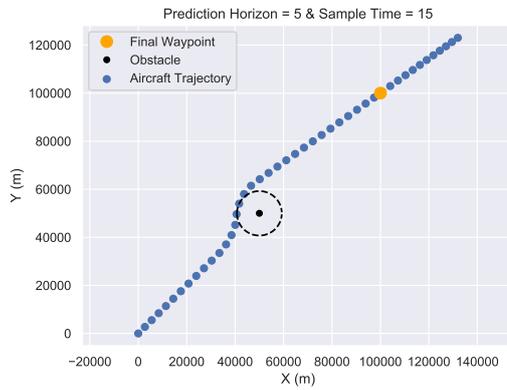


Figure B.7: Aircraft trajectory with static obstacle, with $H_p = 5$ and control sample time = 15 seconds.

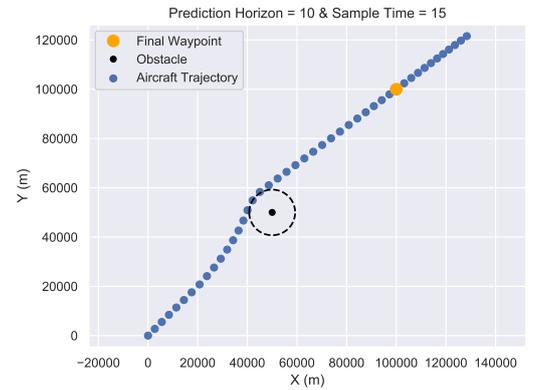


Figure B.8: Aircraft trajectory with static obstacle, with $H_p = 10$ and control sample time = 15 seconds.

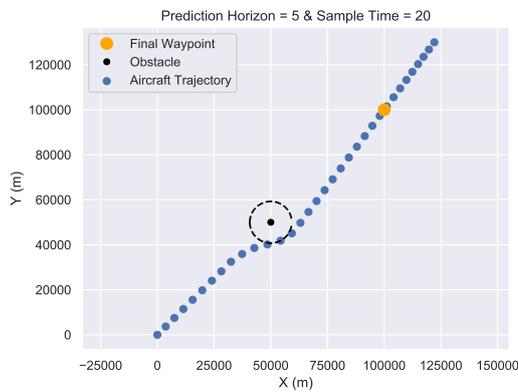


Figure B.9: Aircraft trajectory with static obstacle, with $H_p = 5$ and control sample time = 20 seconds.

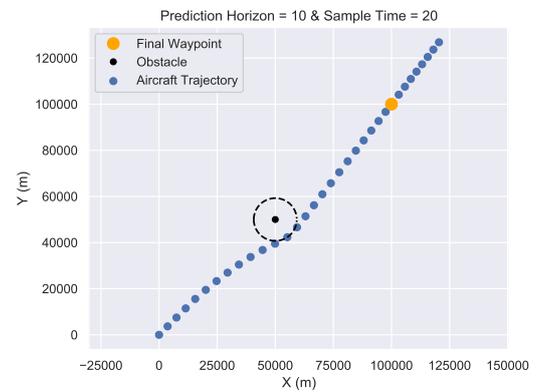


Figure B.10: Aircraft trajectory with static obstacle, with $H_p = 10$ and control sample time = 20 seconds.

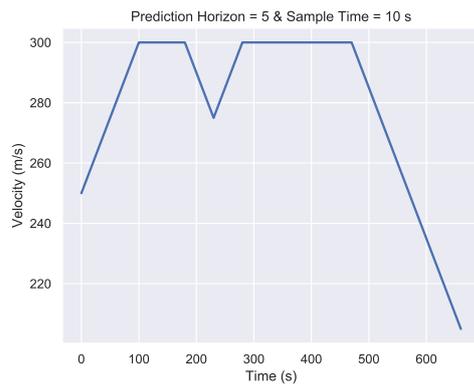


Figure B.11: Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 10 seconds.

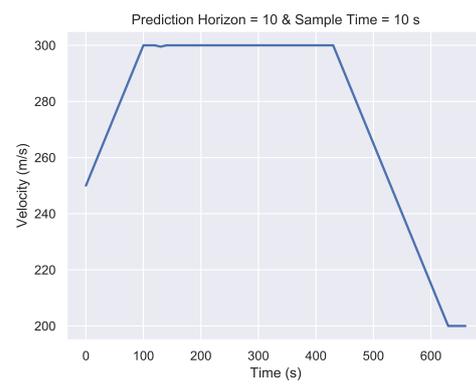


Figure B.12: Aircraft velocity with static obstacle, with $H_p = 20$ and control sample time = 10 seconds.

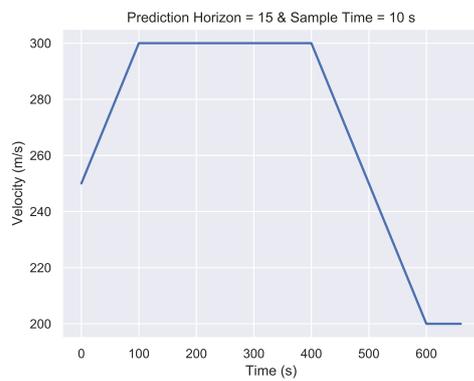


Figure B.13: Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 10 seconds.

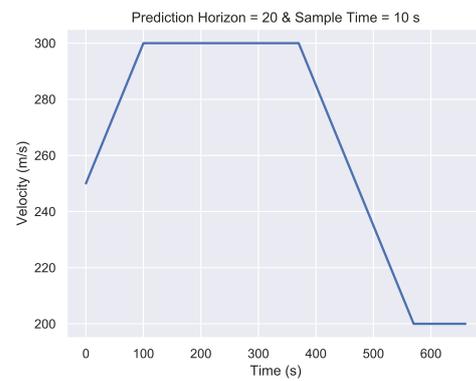


Figure B.14: Aircraft velocity with static obstacle, with $H_p = 20$ and control sample time = 10 seconds.

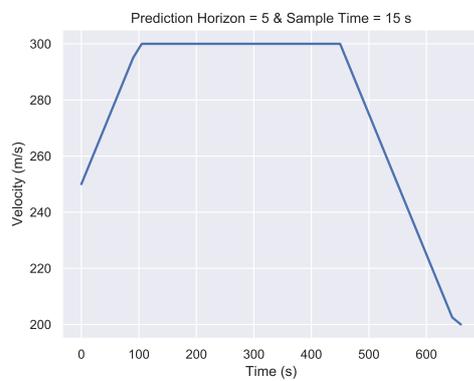


Figure B.15: Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 15 seconds.

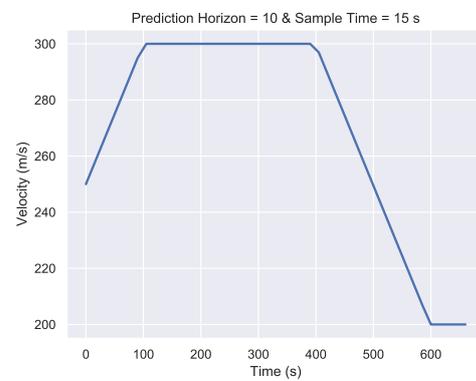


Figure B.16: Aircraft velocity with static obstacle, with $H_p = 20$ and control sample time = 15 seconds.

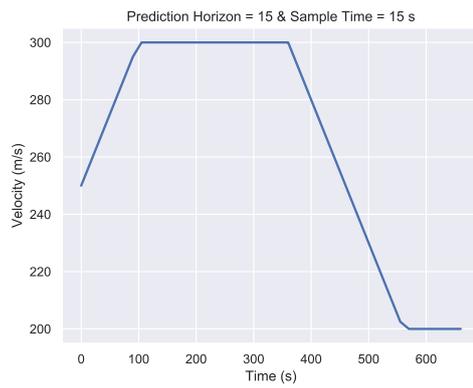


Figure B.17: Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 15 seconds.

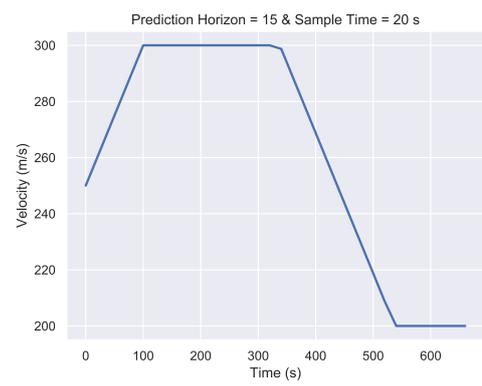


Figure B.18: Aircraft velocity with static obstacle, with $H_p = 15$ and control sample time = 20 seconds.

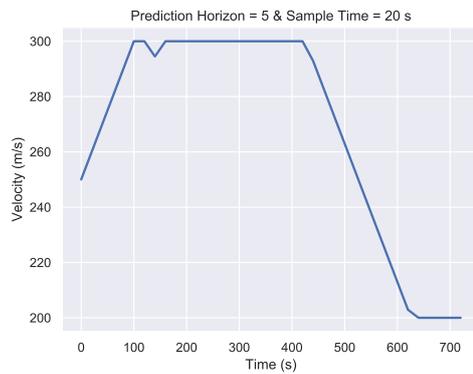


Figure B.19: Aircraft velocity with static obstacle, with $H_p = 5$ and control sample time = 20 seconds.

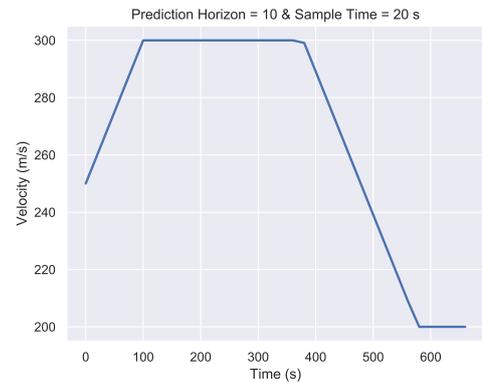


Figure B.20: Aircraft velocity with static obstacle, with $H_p = 10$ and control sample time = 20 seconds.

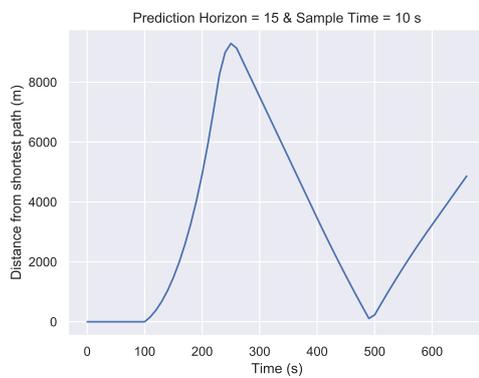


Figure B.21: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 15$ and control sample time = 10 seconds.

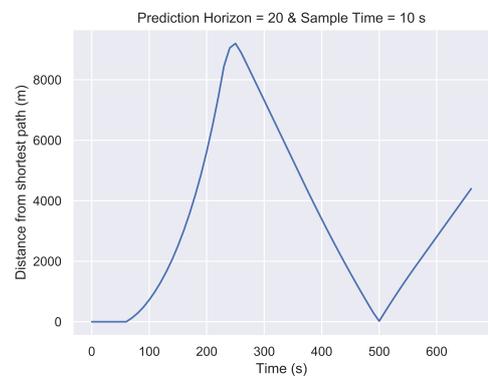


Figure B.22: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 20$ and control sample time = 10 seconds.

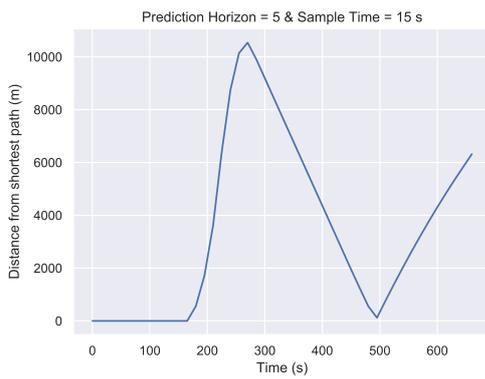


Figure B.23: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 5$ and control sample time = 15 seconds.

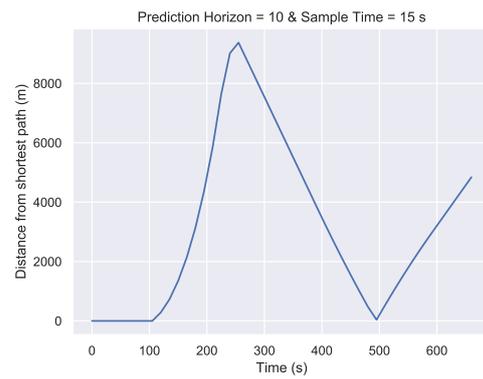


Figure B.24: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 10$ and control sample time = 15 seconds.

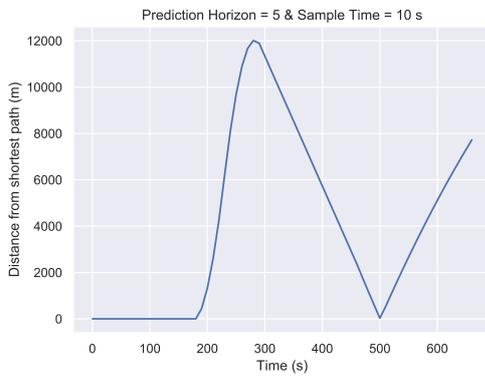


Figure B.25: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 5$ and control sample time = 10 seconds.

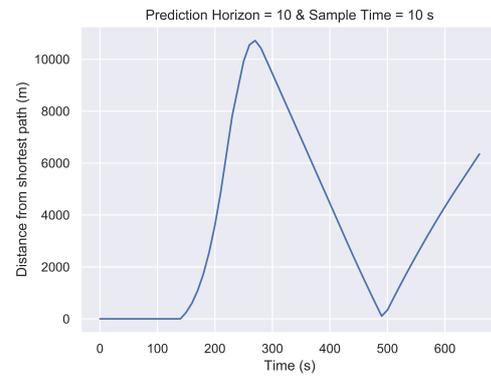


Figure B.26: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 10$ and control sample time = 10 seconds.

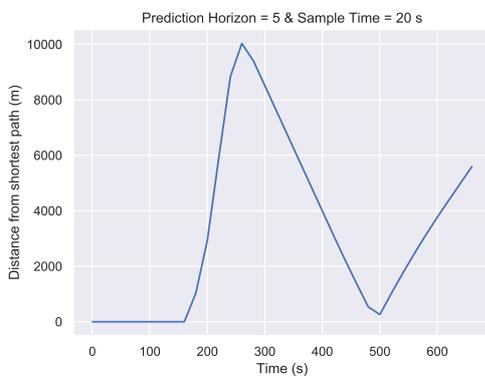


Figure B.27: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 5$ and control sample time = 20 seconds.

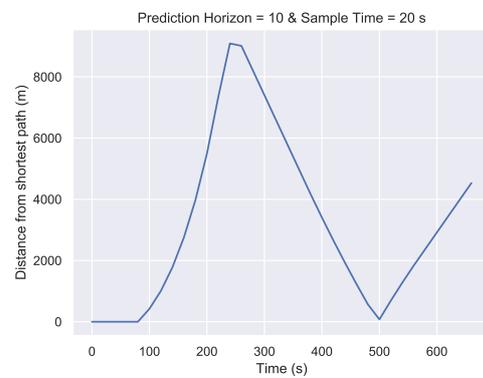


Figure B.28: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 10$ and control sample time = 20 seconds.

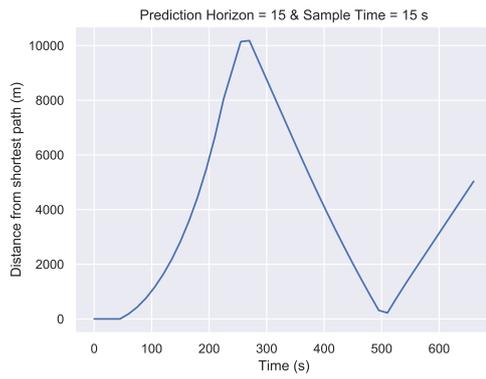


Figure B.29: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 15$ and control sample time = 15 seconds.

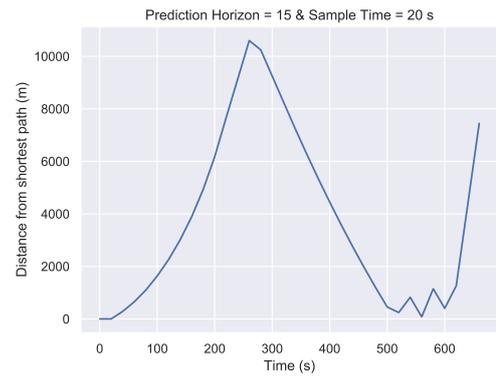


Figure B.30: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 15$ and control sample time = 20 seconds.

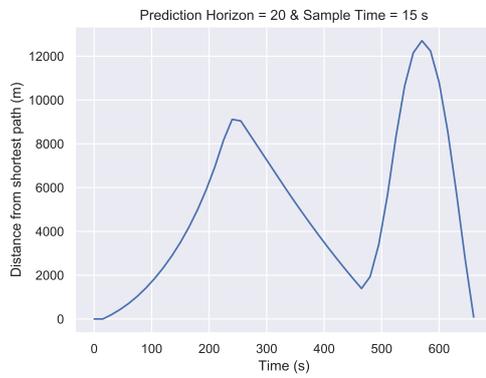


Figure B.31: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 20$ and control sample time = 15 seconds.

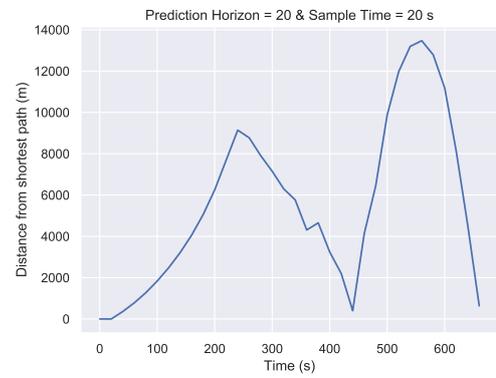


Figure B.32: Aircraft distance from nominal trajectory with static obstacle, with $H_p = 20$ and control sample time = 20 seconds.

C

Appendix C: Boeing 747 Model Parameters

Table C.1: Decentralized and centralized Boeing 747 simulation parameters

B747 Parameters	Decentralized Values	Centralized Values	Unit
S	525.6	525	m^2
ρ	0.4582095	0.5	kg/m^3
M	289600	$0.98 \cdot 289600$	kg
k	0.052	0.052	-
$C_{D,0}$	0.028	0.028	-
N_{eng}	4	4	-
T_{static}	249100	250000	N
A_f	0.2929572527069425	0.3	-
B_f	2.0063275516141754	2.0	-
C_f	0.0303064648693873	0.03	-

Bibliography

- [1] EUROCONTROL. EUROCONTROL Long-Term Forecast: IFR Flight Movements 2010-2030. *Handbook of Local Government Fiscal Health*, 2010.
- [2] Eurocontrol. European Aviation in 2040. Challenges of Growth. page 40, 2018.
- [3] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky. The development of Model Predictive Control in automotive industry: A survey. *Proceedings of the IEEE International Conference on Control Applications*, pages 295–302, 2012.
- [4] Huarong Zheng, Jun Wu, Weimin Wu, and Rudy R. Negenborn. Cooperative Distributed Predictive Control for Collision-Free Vehicle Platoons. *IET Intelligent Transport Systems*, 13(5):816–824, 2019.
- [5] M A Brdys, M Grochowski, T Gminski, K Konarczak, and M Drewa. Hierarchical Predictive Control of Integrated Wastewater Treatment Systems. 16:751–767, 2008.
- [6] Maciej Lawrynczuk, Piotr Marusak, and Piotr Tatjewski. Multilayer and Integrated Structures for Predictive Control and Economic Optimization. 2007.
- [7] L. Ferranti, R. R. Negenborn, T. Keviczky, and J. Alonso-Mora. Coordination of Multiple Vessels Via Distributed Nonlinear Model Predictive Control. *2018 European Control Conference, ECC 2018*, pages 2523–2528, 2018.
- [8] Georgios Chaloulos, Peter Hokayem, and John Lygeros. Distributed Hierarchical MPC for Conflict Resolution in Air Traffic Control. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 44(1 PART 1):1564–1569, 2011.
- [9] Lesley A. Weitz and Xiaoli Bai. Using Model Predictive Control for Trajectory Optimization and to Meet Spacing Objectives. *AIAA Guidance, Navigation, and Control Conference, 2018*, (210039), 2018.
- [10] Xin Min Tang, Ping Chen, and Bo Li. Optimal Air Route Flight Conflict Resolution Based on Receding Horizon Control. *Aerospace Science and Technology*, 50:77–87, 2016.
- [11] K. Bousson. Model predictive control approach to global air collision avoidance. *Aircraft Engineering and Aerospace Technology*, 80(6):605–612, 2008.
- [12] Carlos E. Garcia, David M. Prett, and Manfred Morari. Model Predictive Control : Theory and Practice a Survey. *Automatica*, 25(3):335–338, 1989.
- [13] P.E. Orukpe. Model Predictive Control Fundamentals. 31(2):833–839, 2012.
- [14] E.F. Camacho and C. Bordons. *Model Predictive Control - Second Edition*. 2007.
- [15] J. B. Rawlings, D. Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. 2019.
- [16] Aswin Venkat. Distributed Model Predictive Control: Theory and Applications. *Philosophy*, page 352, 2006.
- [17] Arthur Richards and Jonathan P. How. Model Predictive Control of Vehicle Maneuvers with Guaranteed Completion Time and Robust Feasibility. *Proceedings of the American Control Conference*, 5(April 2014):4034–4040, 2003.
- [18] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained Model Predictive Control: Stability and Optimality. 36(November):1–4, 2011.

- [19] Walid Al-Gherwi, Hector Budman, and Ali Elkamel. Robust Distributed Model Predictive Control: A Review and Recent Developments. *Canadian Journal of Chemical Engineering*, 89(5):1176–1190, 2011.
- [20] D Q Mayne. Control of Constrained Dynamic Systems *. *European Journal of Control*, 7(2-3):87–99, 2001.
- [21] Egidio D’Amato, Massimiliano Mattei, and Immacolata Notaro. Distributed Reactive Model Predictive Control for Collision Avoidance of Unmanned Aerial Vehicles in Civil Airspace. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2019.
- [22] Augustin Degas, Arcady Rantrua, Elsy Kaddoum, Marie-Pierre Gleizes, and Françoise Adreit. Dynamic Collision Avoidance Using Local Cooperative Airplanes Decisions. *CEAS Aeronautical Journal*, (0123456789), 2019.
- [23] Xiao Bing Hu, Shu Fan Wu, and Ju Jiang. On-line Free-Flight Path Optimization Based on Improved Genetic Algorithms. *Engineering Applications of Artificial Intelligence*, 17(8):897–907, 2004.
- [24] Cyril Allignol, Nicolas Barnier, Nicolas Durand, and Jean-Marc Alliot. A New Framework for Solving En-Route Conflicts. *10th USA/Europe Air Traffic Management Research and Development Seminar (ATM 2013)*, 2013(June), 2013.
- [25] Junzi Sun, Joost Ellerbroek, and Jacco M Hoekstra. OpenAP : The Open-Source Aircraft Performance Model and Associated Toolkit. (April), 2019.
- [26] Lars Grüne and Jurgen Pannek. *Nonlinear Model Predictive Control Theory and Algorithms*. 2017.
- [27] Thomas B. Sheridan. Next Generation Air Transportation Systems : Human-Automation Interaction and Organizational Risks. *2nd Resilience Engineering International Symposium*, (December), 2006.
- [28] Rob Ruigrok Dr. Jacco Hoekstra, Mario Valenti Clari, Ronald van Gent. Meeting the Eu Vision 2020 Atc En Route Capacity Goals. (July):1–10, 2003.
- [29] Peter Brooker. SESAR and NextGen: Investing in New Paradigms. *Journal of Navigation*, 61(2):195–208, 2008.
- [30] James K. Kuchar and Lee C. Yang. A Review of Conflict Detection and Resolution Modeling Methods. *IEEE Transactions on Intelligent Transportation Systems*, Vol.I(No.4):pp.179–189, 2000.
- [31] Daniel Delahaye, Nicolas Durand, Jean-Marc Alliot, and Marc Schoenauer. Genetic Algorithms for Air Traffic Control Systems. *INFAUTOM 1996, International Conference on the Optimal Real Time Management of Large Scale Transport Systems*, (September 2016):1–13, 1996.
- [32] Deirdre Bonini, Carole Dupré, and Géraud Granger. How ERASMUS Can Support an Increase in Capacity in 2020. *IMETI 2009 - 2nd International Multi-Conference on Engineering and Technological Innovation, Proceedings*, 3:63–68, 2009.
- [33] J. M. Hoekstra, R. N.H.W. Van Gent, and R. C.J. Ruigrok. Designing for safety: The ‘free flight’ air traffic management concept. *Reliability Engineering and System Safety*, 75(2):215–232, 2002.
- [34] Jerom Maas, E Sunil, Joost Ellerbroek, and Jacco M. Hoekstra. The Effect of Swarming on a Voltage Potential-Based Conflict Resolution Algorithm. *Proceedings of the 7th International Conference on Research in Air Transportation*, (June), 2016.
- [35] Junling Cai and Ning Zhang. Mixed Integer Nonlinear Programming for Aircraft Conflict Avoidance by Applying Velocity and Altitude Changes. *Arabian Journal for Science and Engineering*, 44(10):8893–8903, 2019.
- [36] Sonia Cafieri and Nicolas Durand. Aircraft Deconfliction with Speed Regulation: New Models from Mixed-Integer Optimization. *Journal of Global Optimization*, 58(4):613–629, 2012.
- [37] Sonia Cafieri and David Rey. Maximizing the Number of Conflict-Free Aircraft Using Mixed-Integer Non-linear Programming. *Computers and Operations Research*, 80:147–158, 2017.

- [38] J. C. Hill, J. K. Archibald, W. C. Stirling, and R. L. Frost. A multi-agent system architecture for distributed air traffic control. *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, 3:1936–1946, 2005.
- [39] Yoshiaki Kuwata and Jonathan P. How. Cooperative Distributed Robust Trajectory Optimization Using Receding Horizon MILP. *IEEE Transactions on Control Systems Technology*, 19(2):423–431, 2011.
- [40] Marcus Nolte, Marcel Rose, Torben Stolte, and Markus Maurer. Model Predictive Control Based Trajectory Generation for Autonomous Vehicles-An Architectural Approach. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 798–805, 2017.
- [41] Thomas Prevot, Stephen Shelden, Joey Mercer, Parimal Kopardekar, Everett Palmer, and Vemol Battiste. ATM Concept Integrating Trajectory-Oriented and Airborne Separation Assistance in the Presence of Time-Based Traffic Flow Management. pages 1–12, 2003.
- [42] PLETER Octavian Thor and DONCIU Bogdan. Free Flight vs. Centralized Air Traffic Management. *Incas Bulletin*, 3(4):67–75, 2013.
- [43] Eric Féron To Romaric Breil, Daniel Delahaye, Laurent Lapasset. Multi-agent Systems for Air Traffic Conflicts Resolution by Local Speed Regulation. *7th International Conference on Research in Air Transportation (ICRAT 2016)*, 2016.
- [44] Thomas Prevot, Vernol Battiste, Everett Palmer, and Stephen Shelden. Air Traffic Concept Utilizing 4D Trajectories and Airborne Separation Assistance. *AIAA Guidance Navigation and Control Conference*, 1(August), 2003.
- [45] Robert Graham, Eric Hoffman, Christian Pusch, and Karim Zeghal. Absolute Vs. Relative Navigation: Theoretical Considerations from an ATM Perspective. 2002.
- [46] Mark G. Ballin, Jacco M. Hoekstra, David J. Wing, and Gary W. Lohr. NASA Langley and NLR Research of Distributed Air/Ground Traffic Management. (October), 2002.
- [47] Thomas Prevot, Paul Lee, Todd Callantine, Nancy Smith, and Everett Palmer. Trajectory-Oriented Time-Based Arrival Operations: Results And Recommendations. *FAA/Eurocontrol R&D Seminar*, 2003.
- [48] Fatemeh Asadi and Arthur Richards. Self-Organized Model Predictive Control for Air Traffic Management. *Proceedings of ATACCS 2015 - 5th International Conference on Application and Theory of Automation in Command and Control Systems*, (September):151–159, 2015.
- [49] Fatemeh Asadi and Arthur Richards. Ad Hoc Distributed Model Predictive Control of Air Traffic Management. *IFAC-PapersOnLine*, 28(25):68–73, 2015.
- [50] G. Inalhan, D.M. Stipanovic, and C.J. Tomlin. Decentralized Optimization, with Application to Multiple Aircraft Coordination. pages 1147–1155, 2003.
- [51] Riccardo Scattolini. Architectures for Distributed and Hierarchical Model Predictive Control - A review. *Journal of Process Control*, 19(5):723–731, 2009.
- [52] George J. Pappas, Gerardo Lafferriere, and Shankar Sastry. Hierarchically Consistent Control Systems. *IEEE Transactions on Automatic Control*, 45(6):1144–1160, 2000.
- [53] Piotr Tatjewski. Advanced Control and On-line Process Optimization in Multilayer Structures. 32(July 2007):71–85, 2008.
- [54] Tamás Keviczky, Francesco Borrelli, Kingsley Fregene, Datta Godbole, and Gary J. Balas. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on Control Systems Technology*, 16(1):19–33, 2008.
- [55] Rudy R. Negenborn, Bart De Schutter, and Hans Hellendoorn. Multi-agent Model Predictive Control for Transportation Networks: Serial versus Parallel Schemes. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 11(PART 1):609–614, 2006.

-
- [56] Chao Ming Ying, Srikanth Voorakaranam, and Babu Joseph. Analysis and Performance of the LP-MPC and QP-MPC Cascade Control System. *Proceedings of the American Control Conference*, 2(June):806–810, 1998.
- [57] David Rey, Christophe Rapine, Rémy Fondacci, and Nour Eddin El Faouzi. Subliminal Speed Control in Air Traffic Management: Optimization and Simulation. *Transportation Science*, 50(1):240–262, 2016.