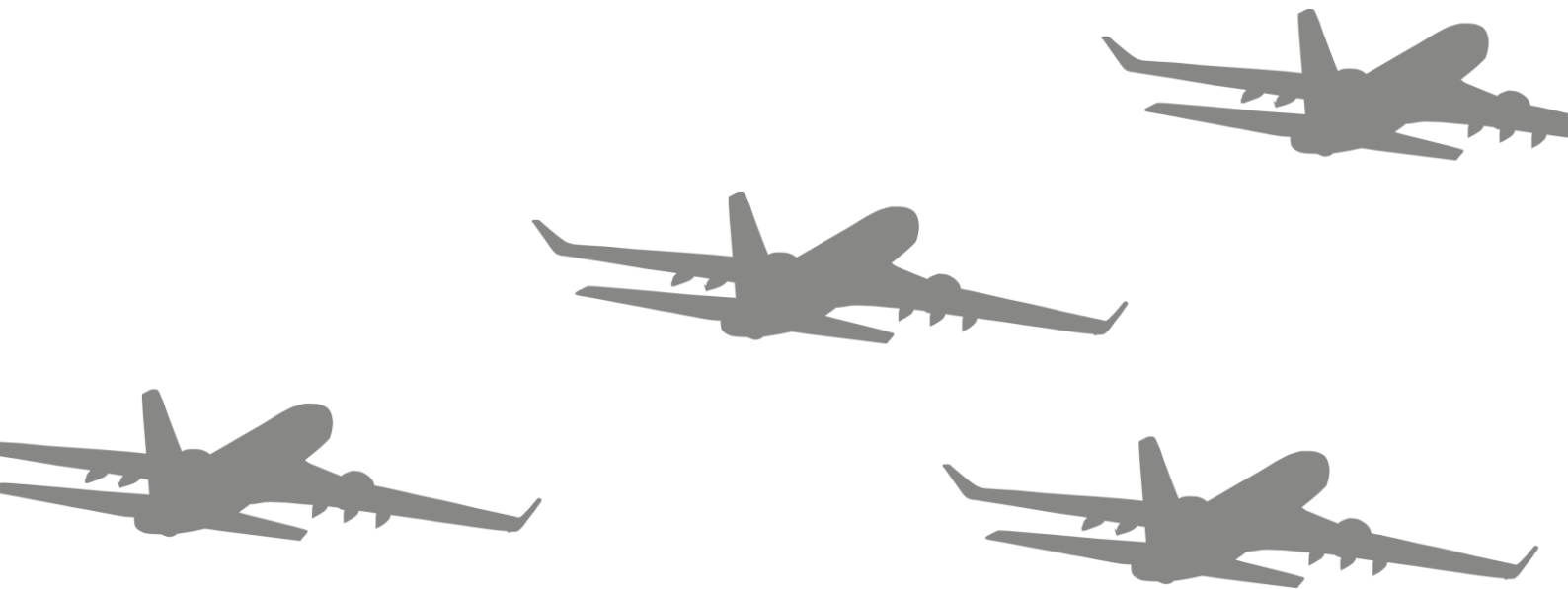


G. Stolwijk

# A Hybrid Optimal Control Approach to Multi-Aircraft Formation Flying

Master of Science Thesis





# A Hybrid Optimal Control Approach to Multi-Aircraft Formation Flying

## Master of Science Thesis

By

G. Stolwijk

in partial fulfilment of the requirements for the degree of

**Master of Science**  
in Aerospace Engineering

at the Delft University of Technology,  
to be defended publicly on Thursday November 30, 2017 at 2:00 PM.

Supervisor:	Dr. ir. S. Hartjes	
Thesis committee:	Dr. ir. S. Hartjes,	TU Delft
	Dr. ir. H.G. Visser	TU Delft
	Dr. ir. E. van Kampen	TU Delft

*This thesis is confidential and cannot be made public until November 30, 2017.*

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



# Executive Summary

With an expected rise in air traffic of 50% [1], new paradigms such as trajectory-based operations (TBO) are in development for air traffic control to enable more efficient aircraft trajectories. Because of this expected development, a great interest is currently shown in methods which can help determine the best trajectory for aircraft to fly given a certain optimization goal. The research field of aircraft trajectory optimization arose as a result, and in general studies in this field are based on optimal control theory using continuous variables. In this thesis, a Hybrid Optimal Control (HOC) framework which can be used in conjunction with existing optimal control software is presented.

Hybrid optimal control theory aims to deal with systems that are both discrete time and continuous time from a mathematical perspective. The hybrid optimal control problem is “to find optimal hybrid – i.e., continuous and discrete – control trajectories such that an integral cost index – typically an integral of a function of the hybrid system state and control input – is minimized subject to the system dynamics, initial, terminal and further equality or inequality constraints” [2]. Much of the complexity of a hybrid optimal control problem depends on how much is known a priori about the switching structure between discrete states.

A case study on multi-aircraft formation flying for civil aviation is chosen as an application to demonstrate the capabilities of the designed HOC method. In formation flight, trailing aircraft can experience a significant reduction in induced drag by ‘surfing’ the upwash of wing tip vortices generated by a leading aircraft. This in turn leads to a significant reduction in fuel burn for the trailing aircraft, making formation flight a more economical and environmentally friendly flying strategy. In previous research [3], the optimization of commercial flights for minimum fuel consumption was studied using multiple-phase trajectory optimization. However, only two-aircraft formations and three-aircraft formations with a fixed phase structure (the sequence in which aircraft join and leave the formation) are considered. By approaching this case as a hybrid optimal control problem, the phase structure can be left open as a discrete optimization parameter, enabling larger and more optimal formation trajectories.

The existing software GPOPS [4], which is already capable of solving multiple-phase optimal control problems, is used as a basis for the developed HOC method. An efficient algorithm for the evaluation of the discrete solution space of the HOC problem, essentially all possible phase structures, is developed through implementation of a branch-and-bound method. In this approach, the discrete solution space is categorized in a tree structure where each tree level represents an additional discrete variable. The algorithm then only evaluates ‘branches’ which show promising characteristics, while ‘chopping’ the other branches. This way, a significant computation time reduction is achieved since a major part of the discrete solution space can be discarded. An initial guess adaptation method which translates trajectories to a correct initial guess for a given phase structure is also developed, as well as an implementation of multi-threaded computing.

The aircraft dynamics are evaluated as a reduced-order point mass model, with the formation flight phases using a single point, multiple mass model to accommodate the presence of multiple aircraft. The induced drag benefit for trailing aircraft is modelled in the aircraft dynamics by implementing a static reduction factor for each trailing aircraft in the relevant formation phases. All phases are connected appropriately using linkage constraints for the state vectors of the linked phases. The used aircraft model characteristics are based on the Boeing 747-400 aircraft.

In several case studies, both a three-aircraft and a four-aircraft transatlantic formation flight scenario are evaluated using the developed HOC tool. The three-aircraft case yields a 4.74% overall reduction in fuel burn, while the four-aircraft case yields an 8.90% reduction. However, when using more pessimistic values for the induced drag reduction (50% less benefit), the fuel burn saving for the four-aircraft case is reduced to 3.48%. The fuel burn reduction also comes at the expense of flight time, since aircraft have to make a detour to join the formation. A three-aircraft case with existing scheduled flights to New York (JFK) is also evaluated, and yields an 11.47% fuel burn saving, a higher value because no significant detours have to be flown. In all experiments, a suboptimality is identified because the aircraft order in the formation is not part of the optimization.

An alternative method for evaluating the fuel burn for the cruise formation phase, by using an analytical Bréguet range equation instead of a dynamic model is also tested. This method yields promising results; no significant deviation is observed in the fuel burn values compared to the point-mass dynamic model. However, the method turns out to be less computationally efficient for GPOPS.

It is found that the developed tool is efficient and functional at evaluating the HOC problem of multi-aircraft formation flight trajectory optimization. Even for the four-aircraft experiments, computation time remains within practical limits at 5 hours. The convergence behavior of GPOPS is identified to be heavily influenced by the accuracy of the adapted initial guess. However, the robustness of the developed HOC tool leaves room for improvement.

To conclude, it is found that the usefulness of a HOC approach is very high for the chosen application of trajectory optimization for multi-aircraft formation flying, since a promising fuel saving potential is identified. Given the number of flights that are performed every day, the impact of saving fuel just by optimizing their trajectories using formation flying can be very high on a global scale, both environmentally and economically.

# Table of Contents

Executive Summary.....	V
List of Figures.....	IX
List of Tables .....	XI
List of Abbreviations.....	XIII
Nomenclature.....	XV
1 Introduction .....	1
1.1 Hybrid Optimal Control .....	1
1.2 Formation Flying in Civil Aviation.....	2
1.3 Optimization software .....	3
1.4 Research Proposal .....	4
1.5 Thesis Structure .....	5
2 Optimization using hybrid optimal control .....	7
2.1 Conventional optimal control and numerical methods review.....	7
2.2 Hybrid Optimal Control Theory .....	13
2.3 Overview of recent aviation research involving hybrid optimal control .....	16
2.4 Analysis of currently common HOC problem-solving techniques .....	18
3 Formation flight and its application for civil aviation .....	27
3.1 Aerodynamic concept .....	27
3.2 Positioning between aircraft in formation.....	28
4 Optimization problem description .....	31
4.1 Problem background.....	31
4.2 Problem statement.....	32
4.3 Optimization goal.....	32
4.4 Application of hybrid optimal control theory.....	33
5 Methodology for trajectory optimization.....	35
5.1 Optimization tool requirements .....	35
5.2 Existing optimal control software as a starting point .....	35
5.3 FORT (GPOPS) working structure and limitations.....	36
5.4 Hybrid Optimal Control approach – Branch-and-bound .....	38

5.5	Outputs and final structure .....	43
6	Case Studies: Experimental set-up .....	45
6.1	Aircraft dynamics model .....	45
6.2	Formation flight set-up .....	47
6.3	Model constraints.....	48
6.4	Aircraft model (Boeing 747-400) .....	52
6.5	Bréguet range equation for formation flight.....	53
6.6	Optimization parameters.....	55
6.7	Assumptions and simplifications for the experiments.....	55
7	Case Studies: Results.....	57
7.1	Experiment 1: Model validation with three-aircraft transatlantic scenario .....	57
7.2	Experiment 2: Four-aircraft transatlantic formation .....	60
7.3	Experiment 3: Sensitivity analysis with less induced drag benefit .....	63
7.4	Experiment 4: Joint-Venture scenario: KLM, Air France, Virgin Atlantic.....	65
7.5	Experiment 5: Breguet vs. non-Breguet for cruise.....	69
8	Discussion .....	71
8.1	Overall evaluation of formation flight experiments.....	71
8.2	Evaluation of the Hybrid Optimal Control approach.....	72
9	Conclusion and Recommendations .....	75
9.1	Conclusion .....	75
9.2	Recommendations .....	77
9.3	Final remarks.....	78
A	Branch-and-bound algorithm flowchart .....	79
B	Initial guess adaptation tool flowchart.....	81
C	Solo flight trajectories (Experiment 1,2,3).....	83
D	Solo flight trajectories (Experiment 4,5).....	85
	References .....	87



# List of Figures

Figure 1: Five civil aircraft flying in formation for demonstration purposes [40] .....	3
Figure 2: Different kinds of systems [5].....	7
Figure 3: Visualization of indirect versus direct methods for solving an optimal control problem numerically [4]. .....	11
Figure 4: Selection process of the optimization method performed by Dr. ir. S. Hartjes [8]. .....	13
Figure 5: (a) Hybrid system. (b). Hybrid control system [12]. .....	14
Figure 6: Example of a branch-and-bound search tree. ....	19
Figure 7: Relation between scaled time $\tau$ and real (unscaled) time $t$ [15].....	23
Figure 8: Visualization of wing tip vortices generated by an aircraft. ....	27
Figure 9: Vertical component of air velocity induced by an aircraft wingtip vortices behind an aircraft [24]......	28
Figure 10: Effect of upwash on thrust required for forward flight [25]......	28
Figure 11: Three common formation shapes [27]. ....	29
Figure 12: Working structure of FORT .....	38
Figure 13: Graphical representation of a phase structure array.....	40
Figure 14: Solution tree for a three-aircraft formation flight case. 0* indicates that phases are still to be added on here further down the tree. ....	41
Figure 15: Hierarchical structure of various HOC tool elements .....	44
Figure 16: Boeing 747-400.....	53
Figure 17: Experiment 1 branch-and-bound evaluated solution tree with cost values per node [ $*10^3$ kg] .....	58
Figure 18: Horizontal trajectory of Experiment 1: solo (dotted lines) vs. formation (solid lines).....	59
Figure 19: Experiment 1 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right). ....	59
Figure 20: Horizontal trajectory of Experiment 2: solo (dotted lines) vs. formation (solid lines).....	61
Figure 21: Experiment 2 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right). ....	61
Figure 22: Horizontal trajectory of Experiment 3: solo (dotted lines) vs. formation (solid lines).....	64
Figure 23: Experiment 3 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right). ....	64
Figure 24: Horizontal trajectory of Experiment 4: solo (dotted lines) vs. formation (solid lines).....	67

Figure 25: Experiment 4 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right). ..... 67

Figure 26: Horizontal trajectory of Experiment 4: solo (dotted lines) vs. formation with Bréguet (solid lines, cruise phase is not modelled)..... 69

Figure 27: Experiment 5 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right). Cruise phase is not modelled. .... 70

# List of Tables

Table 1: Assessment of HOC solving techniques .....	25
Table 2: Description of GPOPS input parameters .....	37
Table 3: Outputs generated by the developed HOC tool .....	43
Table 4: Values for <b><i>cf</i></b> in all experiments unless stated otherwise .....	48
Table 5 Overview of incorporated path constraints .....	51
Table 6: Boeing 747-400 parameters .....	52
Table 7: Flights for Experiment 1.....	57
Table 8: Cost value of all phase structure combinations for Experiment 1.....	58
Table 9 Flights for Experiment 2.....	60
Table 10: Experiment 2 fuel consumption and flight time results, solo vs formation.....	60
Table 11: Values for <b><i>cf</i></b> in Experiment 3.....	63
Table 12: Experiment 3 fuel consumption and flight time results, solo vs formation.....	63
Table 13: Flights for Experiment 4.....	66
Table 14: Experiment 4 fuel consumption and flight time results, solo vs formation.....	66
Table 15: Experiment 5 fuel consumption results .....	69



# List of Abbreviations

ATC	Air Traffic Control
ATM	Air Traffic Management
CPU	Central Processing Unit
FL	Flight Level
GPOPS	General Pseudospectral Optimal control Software
HBVP	Hamilton Boundary-Value Problem
HOC	Hybrid Optimal Control
INTLAB	Interval Laboratory
ISA	International Standard Atmosphere
KKT	Karush-Kuhn-Tucker conditions
LG	Legendre-Gauss
LGL	Legendre-Gauss-Lobatto
LGR	Legendre-Gauss-Radau
MINLP	Mixed Integer Nonlinear Programming
MPBVP	Multipoint Boundary Value Problem
MTOW	Maximum Take-Off Weight
NLP	Nonlinear Programming
OC	Optimal Control
ODE	Ordinary Differential Equation
OEW	Operational Empty Weight
RIOCP	Relaxed Integer Optimal Control Problem
RPM	Radau Pseudospectral Method
SESAR	Single European Sky ATM Research
SNOPT	Sparse Nonlinear Optimization Tool
TBO	Trajectory-Based Operations
TPBVP	Two-Point Boundary Value Problems



# Nomenclature

$\beta$	Relaxed switching function	
$\beta_L$	Lower bound	
$\beta_U$	Upper bound	
$\gamma$	Flight path angle	[rad]
$\gamma$	Specific heat ratio	
$\varepsilon$	Upwash angle	[rad]
$\eta$	Engine thrust setting	
$\lambda$	Longitude	[rad]
$\Lambda$	Adjoint to differential equations	
$\mu$	Roll angle	[rad]
$\nu$	Lagrange multipliers for boundary conditions	
$\rho$	Air density	[kg/m <sup>3</sup> ]
$\tau$	Scaled time	
$\chi$	Heading angle	[rad]
$\phi$	Latitude	[rad]
$\Phi$	Mayer term	
$a$	Speed of sound	[m/s]
$C_{D_0}$	Zero-lift drag coefficient	
$C_0$	Engine performance empirical coefficient	
$C_D$	Total drag coefficient	
$C_L$	Lift coefficient	
$C_M$	Engine performance empirical coefficient	
$c_f$	Induced drag reduction factor	
$C_f$	Induced drag reduction factor array	
$D$	Drag force	[N]
$E$	Specific energy	[J/N]
$g_0$	Gravitational acceleration	[m/s <sup>2</sup> ]
$\mathcal{H}$	Hamiltonian	
$J$	Cost functional	
$K$	Induced drag coefficient	
$L$	Lagrange term	
$M$	Mach number	
$M_{MO}$	Maximum Mach number	
$\dot{m}_{fuel}$	Fuel flow	[kg/s]
$\mathcal{N}$	Branch-and-bound discrete tree node	
$p$	Air pressure	[kg/m <sup>3</sup> ]
$R$	Range / distance flown	[m]
$R_e$	Earth radius	[m]
$S$	Wing surface area	[m <sup>2</sup> ]
$T$	Thrust force	[N]
$T_{max}$	Maximum available thrust	[N]
$T_{min}$	Minimum available thrust	[N]

$t$	Time	[s]
$t_0$	Initial time	[s]
$t_f$	Final time	[s]
$\mathcal{T}$	Branch-and-bound discrete tree	
$u$	Control	
$V$	Velocity	[m/s]
$V_{eff}$	Effective velocity vector	
$V_{min}$	Minimum velocity	[m/s, kts]
$W$	Weight	[N]
$W_0$	Initial weight	[N]
$W_{FC}$	Fuel capacity	[N]
$W_{PL}$	Payload weight	[N]
$W_f$	Final weight	[N]
$W_{max}$	Maximum Take-off Weight	[N]
$W_{min}$	Minimum weight	[N]
$w$	Discrete switching function	
$x$	State	
$z$	Altitude	[m]
$z_{max}$	Maximum altitude	[m]



# 1 Introduction

As both environmental and economic concerns continue to put pressure on the aviation industry, continuous research is being done on how to improve the performance of aircraft in such a way that both the environmental impact and the operational costs are minimized. At the same time, forecasts indicate that by 2035, air traffic in Europe will have increased by 50% compared to 2012, to 14.4 million flights per year [1]. Within the scope of air traffic management, it is expected that operating procedures will be newly developed based on the concept of trajectory-based operations (TBO) [1]. This shift will enable aircraft to fly with a focus on optimal performance since they no longer have to follow relatively inefficient airway paths instructed by air traffic control (ATC). ATC will then only provide restrictions on the trajectories flown by aircraft to ensure maximum safety. This new paradigm means that airspace can be used more flexibly and efficiently, and also that aircraft will be able to fly more efficient trajectories.

Because of this expected development, a great interest is currently shown in methods which can help determine the best trajectory for aircraft to fly given a certain optimization goal. This field of research is called trajectory optimization, and studies on this topic have been performed for several decades. Trajectory optimization studies are usually based on optimal control theory, as it is the only rigorous method which considers all dynamic effects [5]. In optimal control theory applications for aviation, continuous variables are used which describe the state of the aircraft, which evolves along its trajectory through differential-algebraic equations. Several numerical methods for solving optimal control problems for trajectory optimization purposes have been developed, and a survey on these methods is given in [6].

## 1.1 Hybrid Optimal Control

The usefulness of using optimal control theory has been demonstrated numerous times. However, having only continuous variables modeled poses several limitations on the applicability to real-world problems, as many flight trajectories consist of multiple phases and routing elements, each with their own performance objectives and constraints. Traditional optimal control theory cannot incorporate such a trajectory with discrete changes and can thus not optimize for the best possible structure of phases. Therefore, commonly only separate phases are analyzed with optimal control theory individually and then linked together. This however does not give the overall optimal trajectory, because the interaction between different phases is not taken into account. To tackle this, it is attempted to solve trajectory optimization problems by seeing an aircraft as a *hybrid system*, with both continuous and discrete (integer) variables being modeled simultaneously. This approach to trajectory optimization problems is called *Hybrid Optimal Control Theory (HOC)*, and in the last decade, significant strides have been made in research concerned with finding a practical and solvable way to deal with hybrid optimal control problems.

## 1.2 Formation Flying in Civil Aviation

Based on the phenomenon of the formation in which birds tend to fly, which was studied in the 1970s by Lissaman et al. [7], formation flight as a concept for civil aviation to reduce fuel consumption is a field of research where significant knowledge has been obtained by various researchers in the last two decades. The idea is that by letting civil aircraft which share a long enough section of their flight path fly in formation together, the trailing aircraft will experience a significant reduction in fuel consumption through aerodynamic benefits. An additional benefit of this concept is that with aircraft flying closer together in a formation airspace is used more efficiently, allowing for a larger number of flights. With the expected arrival of TBO and the associated higher individual freedom of aircraft routing, civil aviation formation flying becomes more feasible, which leads to a higher interest in the development of solutions for practical implementation.

In a thesis by M.E.G. van Hellenberg Hubar [3], the optimization of commercial flights for minimum fuel consumption was studied using multiple-phase trajectory optimization. A tool was developed which is capable of finding the optimal trajectories of multiple aircraft which join in formation with the objective of minimizing the total fuel consumption of these aircraft. Using this tool, it was successfully demonstrated with several experiments that formation flight could lead to significant fuel savings on transatlantic flights while keeping the increase in flight time within limits. In most of the experiments performed in this thesis, a case was simulated with two identical aircraft making a rendezvous to form a formation at an optimal point along their way to their destination and also going out of formation to fly individually at the most optimal point. Also, an experiment was performed where instead of two, three aircraft would join in formation, which is potentially beneficial since then two aircraft can benefit from a reduction in induced drag, even though there remains only one aircraft which does not benefit from the formation (the leading aircraft). However, finding an optimization approach to finding the optimal formation composition and scheduling was considered beyond the scope of the thesis, resulting in arbitrarily chosen values for the experiments. It was recommended that further investigation would be done regarding the possibility of flying with three or more aircraft of different types in formation, since such scenarios could heavily influence the operational performance of formation flying.

This is a research gap where a hybrid optimal control approach could make a direct impact to the solvability of such problems, making it a suitable application for which a HOC software tool could be developed.



Figure 1: Five civil aircraft flying in formation for demonstration purposes [40]

### 1.3 Optimization software

Trajectory optimization problems, especially concerning aviation, are in general computationally intensive problems to solve. Therefore, choosing a suitable numerical solution which can deal with such type of problems in an efficient manner is of high importance for a successful outcome. There are several software packages available which already integrate one or more methods to solve optimal control problems, and using one of these software packages as a foundation for further research is recommended to keep the size of a hybrid optimal control-focused research project within limits. Having a user interface readily available to solve problems makes for a much easier process of testing multiple optimization cases.

GPOPS (General Pseudospectral Optimization Software) [4] is an optimization tool integrated in the versatile MATLAB environment, and has been made available to use as a foundation for hybrid optimal control development. GPOPS is already capable of solving multiple-phase optimal control problems, where phases are linked as defined by the user such that the total outcome is valid for the user's mission requirements. Since GPOPS is already capable of solving multi-phase optimal control problems, where the switching times between phases can be left open to be solved for, this software package already incorporates hybrid functionality to a certain degree. In its current state, GPOPS is however not able to solve problems where the sequence and number of phases is also left open. Theoretically, a solver would then be able to pick the best scenario described as a set of switches and their respective times of occurrence. The aim of this research project is to incorporate a degree of combinatorial complexity to the functionality of GPOPS, such that an overall more optimal solution to hybrid problems can be obtained.

## 1.4 Research Proposal

In this thesis, an in-depth study is presented covering the development of a software optimization tool based on hybrid optimal control principles. The research goal is to develop a working numerical implementation of hybrid optimal control theory as an extension to existing optimal control software, designed to solve a civil aviation formation flying optimization problem. It is proposed to develop a HOC optimization algorithm which can extend on the thesis research performed by [3] by finding the optimal formation strategy when the number of aircraft is larger than two. Using this tool, multiple scenarios with aircraft flying a similar route can be simulated. The sequence and timing of aircraft forming one or possibly more than one formation is to be found by the HOC algorithm in combination with GPOPS with minimal overall fuel consumption being the objective.

The **research question** can be defined as follows:

*“Can the existing optimal control software tool GPOPS be extended with functionality that is capable of solving the hybrid optimal control problem of formation flying trajectory optimization with 3 or more aircraft, and can this be both verified and demonstrated with case studies?”*

Sub-questions to this main research question are:

- 1) How, and using which approach should the hybrid functionality be implemented?
  - a) To what extent must GPOPS be modified to adapt to HOC techniques?
  - b) How are the trade-off criteria for the choice of hybrid approach defined?
  - c) How can the implementation be made such that it is suitable for use and development in further research?
  - d) Is the complexity of the chosen approach manageable within the timeframe of an MSc thesis?
- 2) Is the chosen and developed implementation valid and verifiable?
  - a) Are there any limitations to the chosen approach which limit the applicability?
  - b) Can results of an existing research case study be reproduced with the developed algorithm?
  - c) Does the algorithm provide the output that is actually needed?
- 3) What difference does using hybrid control software make compared to conventional optimal control techniques?
  - a) Using case studies which can be used with regular OC and HOC, is the difference in cost functional significant?
  - b) Will the additional complexity added by HOC be justifiable given the achieved results?

Hybrid optimal control theory should be regarded as the cornerstone behind the upcoming MSc Thesis project. The **objective** of the project will be:

*“To have a working aircraft trajectory optimization model based on GPOPS which is extended with the working principle of hybrid optimal control theory in such a way that it enables trajectory optimization for a formation flight of 3 or more aircraft. At least one case study which verifies the correct functionality of the code, as well as at least one case study which demonstrates the capabilities of the code for multi-aircraft formation flight should be executed and analyzed. “*

Sub-goals to this project objective are then defined as follows:

- 1) To find a suitable numerical approach to HOC from (combinations of) work published in existing literature
- 2) To analyze and understand the working principles behind GPOPS and identify which elements to modify or extend
- 3) To build a HOC algorithm based on acquired knowledge
- 4) To test the algorithm using simple case studies and iterate until outcomes are as desired
- 5) To use the newly built algorithm to solve an aviation trajectory optimization problem not done before, e.g. formation determination in formation flight trajectories.
- 6) To draw conclusions about the effectiveness of the chosen HOC method as well as the applicability to the chosen case study

## 1.5 Thesis Structure

The scope of this thesis covers the development of a software tool based on hybrid optimal control for the trajectory optimization of civil aviation formation flying, as well as several numerical case studies which are performed using this tool and analyzed. Chapter 2 discusses the concepts behind conventional optimal control theory and hybrid optimal control theory. Also, an analysis of currently available methods for dealing with hybrid optimal control problems and an overview of previous research on this topic is presented here. Chapter 3 covers the concept of formation flying and the potential aerodynamic benefits which can be applied to civil aviation. Chapter 4 discusses the optimization problem which arises when planning the trajectories for multiple aircraft which are candidates to join each other in formation, as well as the nature of this problem as a hybrid optimal control problem. Next, in Chapter 5 the methodology for solving this optimization problem by developing a hybrid optimal control-based optimization tool is covered. Chapter 6 then discusses the experimental set-up for the case studies, including the aircraft dynamic modelling in formation flight and solo. The results of the performed case studies are presented in Chapter 7, and also assessed per case study individually. In Chapter 8, a more general discussion about the trajectory optimization results as well as the performance of the developed hybrid optimal control tool is presented, and based on this, conclusions regarding this thesis project and recommendations for future research are presented in Chapter 9.



## 2 Optimization using hybrid optimal control

### 2.1 Conventional optimal control and numerical methods review

Because Hybrid optimal control theory can be seen as an extension to traditional optimal control theory and trajectory optimization, it is essential that this theory is also understood to the extent that the major principles are clear. Additionally, having sufficient knowledge about the numerical methods used in the available software packages which utilize optimal control theory is also considered useful.

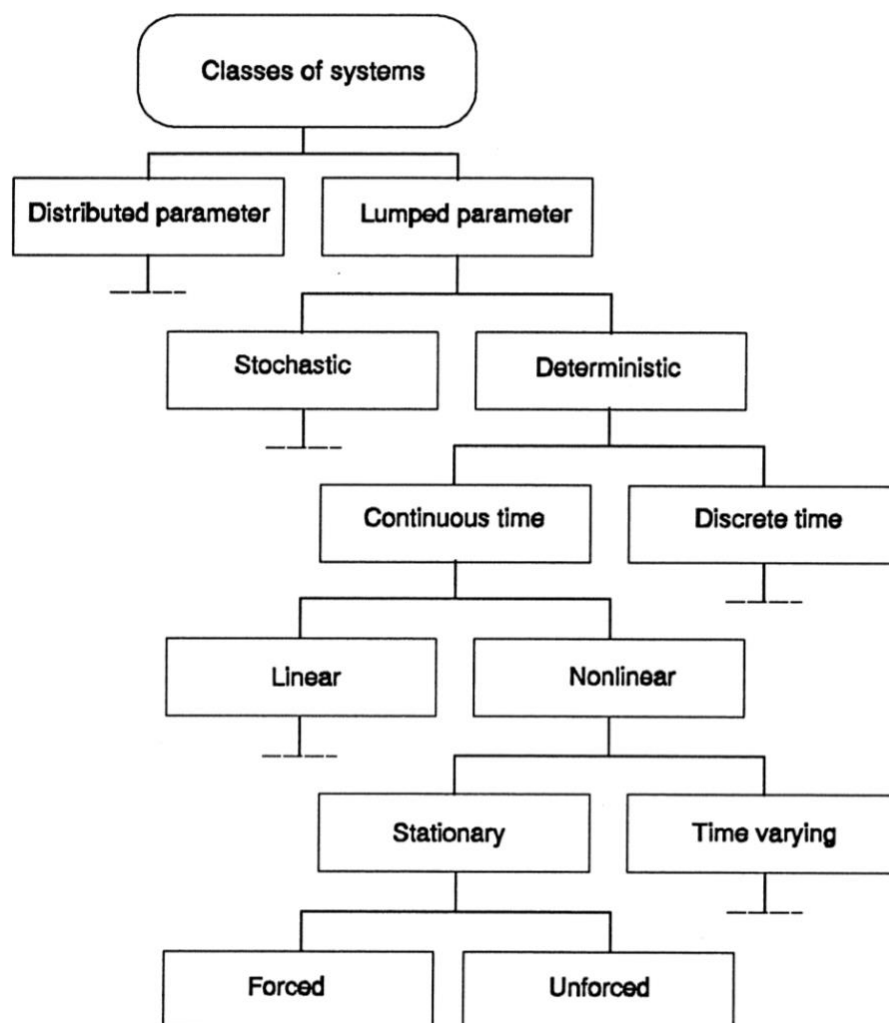


Figure 2: Different kinds of systems [5]

### 2.1.1 Statement of the Optimal Control Problem

The theme which covers the entire scope of this research is *optimization*. Real world optimization problems can be modeled mathematically using a *systems approach*. Systems can be classified according to the types of equations used to describe them. This classification is presented in Figure 2. Several approaches exist towards dealing with optimization problems, however in this research a major focus lies on the approach using *optimal control theory*, which is useful for dealing with *continuous time, nonlinear systems*. The end goal of this research is to successfully implement *hybrid optimal control theory*, which aims to deal with systems that are both *discrete time and continuous time* from a mathematical perspective.

Continuous time systems are often *discretized*: an approximation of the continuous time model is made using discrete-time system models to make the optimization problem solvable using numerical methods (i.e. using computers).

Systems can be determined by a descriptive set of parameters, called the *state* of the system. Each of these parameters is called a *state variable* and is denoted as  $x_i, i=1,2,\dots,n$ . The state of systems is generally not steady, but it evolves through time, and this evolution often depends on the *inputs* to the system. These inputs are mathematically represented as *control variables*, and denoted as  $u_k, k=1,2,\dots,m$ . The (physical) behavior of the system is then described by ordinary differential equations in the following form:

$$\dot{\underline{x}} = \underline{f}[\underline{x}(t), \underline{u}(t), t] \quad (2.1)$$

The *trajectory*  $\underline{x}$  of a system is then defined as a description of the evolution of the state between the *initial* and *final time*. The *control*  $\underline{u}$  of a system is the evolution of the applied control inputs between the initial and final time. Controls are usually constrained to certain (physical) limits i.e.  $\underline{u} \in U$ . Generally, in an optimization problem, one aims to optimize the trajectory. Before one can do this, it is necessary to define a *performance measure* which describes how optimal a trajectory is with respect to the ultimate goal of the optimization problem. Then, an *optimal control* is a control which *minimizes* the performance measure.

The performance measure  $J$  is expressed as follows:

$$J = \Phi[\underline{x}(t_f), t_f] + \int_{t_0}^{t_f} L[\underline{x}(t), \underline{u}(t), t] dt \quad (2.2)$$

Where  $\Phi$  represents the *endpoint cost (Mayer term)*, and  $L$  represents the *running cost (Lagrange term)*.  $J$  is also referred to as the *cost functional*.



Finally, the “**Optimal Control Problem**” can be stated as follows:

“From among all admissible control functions  $\underline{u} \in U$ , find that one which minimizes  $J$  of Eq. (2.2) subject to the dynamic constraints of Eq. (2.1) and all initial and terminal boundary conditions that may be specified.” [5]

This type of optimal control formulation (with both a Mayer and a Lagrange term) is also called the *Continuous Bolza Problem*. The optimal control function that is found is denoted  $\underline{u}^*$  and the respective optimal trajectory is then denoted as  $\underline{x}^*$ .

### 2.1.2 First-Order Optimality Conditions

Now that the optimal control problem has been defined, the next step is to assess how to solve the problem. This procedure is concisely summarized in [8]. To do this, an augmented cost functional can be constructed where the dynamic constraints (Eq. (2.1)) as well as the boundary conditions ( $\phi(x(t_0), t_0, x(t_f), t_f) = 0$ ) are adjoined to the original cost functional as follows:

$$J_a = \Phi - \nu^T \phi + \int_{t_0}^{t_f} [L + \lambda^T (f - \dot{x})] dt \quad (2.3)$$

Where  $\nu$  are Lagrange multipliers for the boundary conditions, and  $\lambda$  are the adjoints to the differential equations. The *Hamiltonian* is defined as the following element of the augmented cost functional

$$\mathcal{H} = L + \lambda^T f \quad (2.4)$$

One aims to find the minimal solution to Eq. (2.3), and for this *Pontryagin’s Minimum Principle* can be applied [9]. This principle states that the gradient of the cost functional in any direction equals zero **if**  $x^*$  is a minimal solution:

$$\delta J(x^*, \delta x) = 0 \quad \forall \text{ admissible } \delta x \quad (2.5)$$

Further application of this principle to Eq. (2.3), leads to a set of conditions which, when satisfied, indicate a *candidate* optimal solution. These conditions are the *First-Order Optimality Conditions* and together with the existing constraints they form the *Hamiltonian Boundary-Value Problem (HBVP)*. They are defined as follows:

$$\dot{x} = f(x(t), u(t), t) \quad (2.6)$$

$$\phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = 0 \quad (2.7)$$

$$\lambda(t_0) = - \left[ \frac{\partial \Phi}{\partial \mathbf{x}(t_0)} \right]^T + \left[ \frac{\partial \phi}{\partial \mathbf{x}(t_0)} \right]^T \mathbf{v} \quad (2.8)$$

$$\lambda(t_f) = - \left[ \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} \right]^T + \left[ \frac{\partial \phi}{\partial \mathbf{x}(t_f)} \right]^T \mathbf{v} \quad (2.9)$$

$$\mathcal{H}(t_0) = \frac{\partial \Phi}{\partial t_0} - \mathbf{v}^T \frac{\partial \phi}{\partial t_0} \quad (2.10)$$

$$\mathcal{H}(t_f) = \frac{\partial \Phi}{\partial t_f} - \mathbf{v}^T \frac{\partial \phi}{\partial t_f} \quad (2.11)$$

$$\dot{\lambda} = - \left[ \frac{\partial \mathcal{H}}{\partial \mathbf{x}} \right]^T \quad (2.12)$$

$$\left[ \frac{\partial \mathcal{H}}{\partial \mathbf{u}} \right]^T = 0 \quad (2.13)$$

Except for very simple problems, it is impossible to solve the HBVP analytically. Therefore, numerical methods are used to find an optimal solution.

### 2.1.3 Numerical methods for optimal control problems: Direct vs. indirect

Generally, numerical methods for finding optimal solutions can be divided into two categories: *indirect* and *direct* methods. The difference between these methods is the route which is taken to get to the final solution. Whereas indirect methods aim to solve the HBVP of Section 2.1.2 first and then discretize the result to be processed numerically (*optimize then discretize*), direct methods make a discrete approximation of the ODE's, and then compute an optimal solution from this as a Nonlinear Programming (NLP) problem (*discretize then optimize*) [6], [8], [10]. A visualization on how these two approaches will lead to the same result is shown in Figure 3:

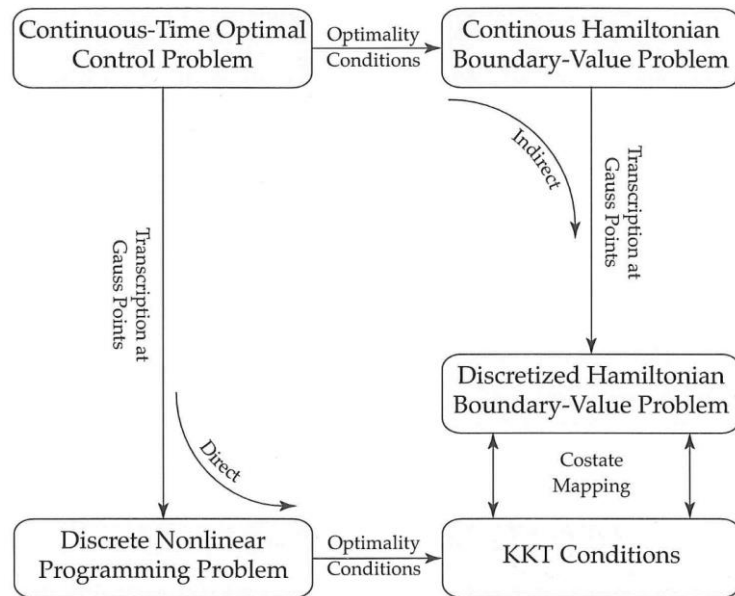


Figure 3: Visualization of indirect versus direct methods for solving an optimal control problem numerically [4].

Naturally, the question then arises which of the two approaches is the better one to take for the types of problems that are dealt with here.

O. von Stryk and R. Bulirsch published a paper in the Annals of Operations Research on direct and indirect methods for trajectory optimization in 1992 [11]. It is mentioned that indirect methods have the advantage that very accurate results can be obtained using multiple shooting. However, several drawbacks are also mentioned:

- A very good initial approximation of the optimal trajectory is needed
- Deriving the adjoint differential equations is a daunting task
- The switching structure of the constraints must be known a priori (or guessed)

These observations were confirmed by John T. Betts [10], and in this textbook another comment was that the robustness of this method can be questioned: The adjoint variables  $\lambda$  must be guessed, which is unintuitive since they are not a physical quantity, and on top of this, the sensitivity of the numerical solution to these guesses is so high that a slightly poor guess of the initial conditions can lead to very unexpected trajectories in the state space. Another practical problem which is mentioned is the fact that whenever even a small change in the constraints or dynamics of the problem needs to be made, the entire derivation must be redone.

The potential of a direct method was recognized in 1992 by von Stryk and Bulirsch [11], as they found that it was much easier to achieve a convergent result, even when no initial information was given about trajectories. The drawbacks that hold for indirect methods do not hold for direct methods. However, they were not entirely satisfied with the accuracy of the results compared to the exact solution obtained from indirect methods. However, given the

drastic developments in NLP solving techniques and the availability of more computing power, it is expected that this disadvantage is much less relevant today. This expectation is also confirmed in [10], given that this book dismisses indirect methods in favor of direct methods because of the mentioned drawbacks for indirect methods.

#### 2.1.4 Shooting versus collocation

When using direct methods, it is necessary to perform a transcription of the optimal control problems into (finite dimensional) NLPs by a parametrization of the control variable  $u$ . There are two strategies for this transcription, being *direct shooting* and *direct collocation*. M. Buss et al. [2] discusses these two strategies, and although there are advantages and disadvantages to either strategy, generally the direct collocation approach is favored due to the fact that it only satisfies the equations of motion at a successful termination of the optimization procedure, resulting in higher accuracy. Also, in [8], it is mentioned that direct collocation allows for a complex set of path constraints to be imposed, which is likely to be necessary in the scenarios that will be tested.

#### 2.1.5 Further considerations

Within collocation methods, a choice can be subsequently made between *global* and *local collocation*, both of which use a different approach to approximate the state and control variables. According to [8], *global collocation*, and then specifically *Pseudospectral collocation* offers the greatest accuracy, as they are based on a form of Gaussian quadrature to integrate the system dynamics. Additionally, the problem size can remain within limits since the number of discretization points required is lower.

There are three types of Gaussian quadrature which can be used: *Legendre-Gauss (LG)*, *Legendre-Gauss-Lobatto (LGL)* and *Legendre-Gauss-Radau (LGR)* quadrature rules. In [8], it is mentioned that LGR is probably the most useful type, as it shows the best convergence characteristics and also includes one boundary point, which is favorable for problems with either an initial or final time fixed. However, different applications might favor other quadrature rules, so it is worthwhile to study this further when an application is built.

A decision tree which summarizes the process of breaking down the numerical methods to solve optimal control problems was presented in [8], and since it clearly shows the core decision process, it is presented in Figure 4:

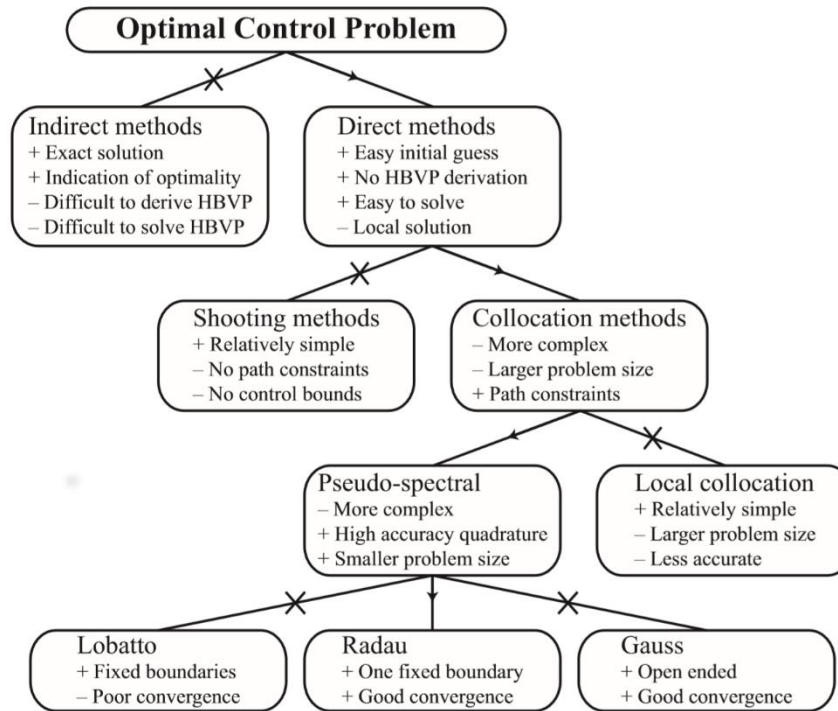


Figure 4: Selection process of the optimization method performed by Dr. ir. S. Hartjes [8].

## 2.2 Hybrid Optimal Control Theory

‘Hybrid’ is a term which can be used very flexibly in many (research) fields. In this section, it is described what is understood to be the definition of hybrid optimal control based on existing research papers. From this, a conclusion will follow on how this term will be defined for the thesis project, such that there will be no more confusion regarding this matter.

### 2.2.1 Definition of the hybrid optimal control Problem

According to Branicky et al. [12], at the base of a HOC problem is a *hybrid system* - the actual system to be controlled -, which is defined as a system which “involves both continuous-valued and discrete-valued variables”. Further, four types of discrete phenomena are classified which can occur in a hybrid system, being *Autonomous Switching*, *Autonomous Impulses*, *Controlled Switching* and *Controlled Impulses*. This work has thus presented a more generalized and complete definition of what makes a system a “hybrid” system.

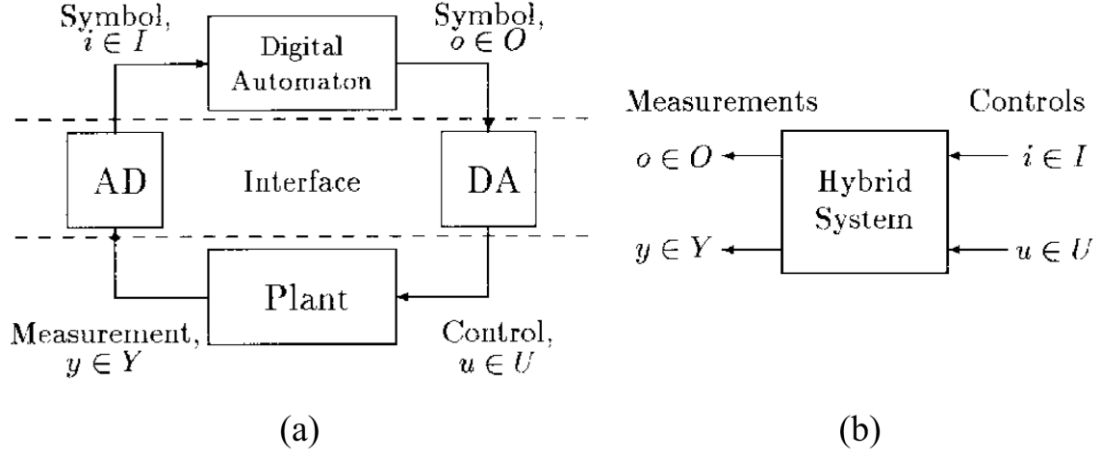


Figure 5: (a) Hybrid system. (b). Hybrid control system [12].

The definition of a **hybrid optimal control problem** given in [2] is as follows:

“The hybrid optimal control problem is to find optimal hybrid – i.e., continuous  $\mathbf{u}$  and discrete  $\mathbf{v}$  – control trajectories such that an integral cost index – typically an integral of a function of the hybrid system state and control input – is minimized subject to the system dynamics, initial, terminal and further equality or inequality constraints.”

Or, as a mathematical definition, “it is defined as the minimization of the hybrid cost index  $J$ :

$$\min_{\mathbf{u}, \mathbf{v}} J(\mathbf{u}, \mathbf{v}) = \Theta + \int_{t_a}^{t_e} \psi(\mathbf{x}, \mathbf{u}, \mathbf{q}, \mathbf{v}, t) dt \quad (2.14)$$

Subject to:

$$\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{q}, \mathbf{v}, t) \quad \text{if} \quad s_j(\mathbf{x}, \mathbf{u}, \mathbf{q}, \mathbf{v}, t) \neq 0 \quad j = 1, \dots, n_s \quad (2.15)$$

$$\begin{bmatrix} \mathbf{x}(t_i^+) \\ \mathbf{q}(t_i^+) \end{bmatrix} = \phi_j(\mathbf{x}, \mathbf{u}, \mathbf{q}, \mathbf{v}, t_i^-) \quad \text{if} \quad s_j(\mathbf{x}, \mathbf{u}, \mathbf{q}, \mathbf{v}, t) = 0 \quad j = 1, \dots, n_s \quad (2.16)$$

$$\mathbf{u}(t) \in \mathcal{U} \subset \mathbb{R}^{n_u}, \mathbf{v}(t) \in \mathcal{V} \subset \mathbb{Z}^{n_v}, \mathbf{x}(t) \in \mathcal{X} \subset \mathbb{R}^{n_x}, \mathbf{q}(t) \in \mathcal{Q} \subset \mathbb{Z}^{n_q}, \quad (2.17)$$

$$\forall t \in [t_a, t_e]$$

$$0 \leq \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{q}, \mathbf{v}, t), t \in [t_a, t_e] \text{ inequality constraints}, \quad (2.18)$$

$$\mathbf{x}(t_a) = \mathbf{x}_a, \mathbf{q}(t_a) = \mathbf{q}_a \text{ initial conditions}, \quad (2.19)$$

$$\mathbf{x}(t_e) = \mathbf{x}_e, \mathbf{q}(t_e) = \mathbf{q}_e \text{ terminal conditions}, \quad (2.20)$$

Where the initial and final times  $t_a, t_e$  are free or fixed,  $s_j$  are the  $n_s$  switching functions and  $\phi_j$  denotes the explicit phase transition conditions (jump maps) occurring at the zeros of one of the switching functions. The Mayer type part  $\theta$  of the performance index is a general function of the phase transition times (events)  $t_i, i = 0, \dots, N$ , of the continuous  $\mathbf{x}(t_i^-), \mathbf{x}(t_i^+)$  and discrete states  $\mathbf{q}(t_i^-), \mathbf{q}(t_i^+)$  just before and just after the transition event written as:

$$\theta := \theta[\mathbf{x}(t_0^-), \mathbf{x}(t_0^+), \dots, \mathbf{x}(t_N^-), \mathbf{x}(t_N^+); \mathbf{q}(t_0^-), \mathbf{q}(t_0^+), \dots, \mathbf{q}(t_N^-), \mathbf{q}(t_N^+); t_0, \dots, t_N] \quad (2.21)$$

Here,  $t_a = t_0, t_e = t_N$  is assumed while the number of phases  $N$  may be given or free. The integrand  $\psi$  is a real-valued function of the continuous/discrete state and control variables and of time.

The minimization of (2.14) is subject to the initial and terminal conditions (2.19), (2.20) admissible values for the continuous/discrete control variables (2.17), and inequality constraints (2.18). Obviously, valid hybrid optimal trajectories have to obey the differential equations (2.15) and the phase transition equations (2.16) of the discrete aspect. The optimization parameters to be determined are the continuous  $\mathbf{u}(t)$  and discrete control input trajectories  $\mathbf{v}(t)$  and all, some or none of the phase transition times." [2]

The above presented mathematical definition can be seen as an extended version of the traditional optimal control problem such as described in Section 2.1.1.

## 2.2.2 Complexity of hybrid systems

In (Buss et al., 2000) [2], it is pointed out that the level of complexity of hybrid systems can vary on several levels. Conventionally, optimal control problems have been presented in the form of two- or multi-point boundary value problems (TPBVP or MPBVP), often extended to cases with inequality constraints on the state and control variables. Later, this was applied to multi-phase problems, where discontinuities exist at intermediate time points. These time points are denoted as *switching times*, and each phase has its own dynamics. Control problems

with multiple phases are considered “hybrid” problems by some, since discrete changes in the state occur when the model switches between phases. However, to solve these multi-phase problems, as the assumption is made that the sequence of these phases (switching structure) is known. This assumption is a major limiting factor for many real world multiphase problems, since in many dynamical systems with both continuous and discrete variables, this switching sequence is not known and assuming a certain sequence will likely lead to a suboptimal overall solution. In the most generic form of hybrid optimal control problems, neither the sequence, nor the times nor the number of the switches is available. To quote, “It is the intrinsic combinatorial complexity, in addition to the nonlinearity of the continuous optimal control problem that forms the challenges in the theoretical and numerical solution of hybrid optimal control problems.”

To conclude, it can be said that the definition of the general class of hybrid optimal control problems is consistently agreed upon. All systems which involve both continuous and discrete input and state variables can be considered hybrid systems, and all optimization problems that aim to find the optimal trajectory of these states using an optimal set of inputs can be considered hybrid optimal control problems. However, within this definition there is a lot of variation possible in the level of complexity of these problems. Much of this complexity depends on how much is known a priori about the switching structure between discrete states. For example, if the switching sequence and switching times are already known, the hybrid multiphase problem is not much more complex than a conventional trajectory optimization problem. It is in this case only necessary to solve the phases individually using conventional methods and then link them together with linkage constraints. If the switching times are unknown, the complexity is increased but still solvable, but if also the sequence of the switches – or even the number of switches – is also still to be determined, the intrinsic combinatorial complexity becomes of a much higher order. It is this latter category of hybrid optimal control problems, where the discrete part is to be determined almost completely, which requires additional methods to be implemented before it can be solved. This is however also the category where the most promising improvements can be made in trajectory optimization. The further aim of this thesis project is to address this generic form of hybrid optimal control problems with simplifications only made when it has no significant effect on the applicability of the method in case studies.

## 2.3 Overview of recent aviation research involving hybrid optimal control

Research on hybrid optimal control theory applications for aviation operations has also been done. The field of trajectory optimization is especially relevant, and thus the prospect of using HOC in this field is being heavily investigated. Especially since 2010, one can observe a significant increase in publications discussing aerospace applications of hybrid optimal control models.

### 2.3.1 Optimal Control of Hybrid Systems in Air Traffic Applications (M. Kamgarpour, 2010)

For a PhD Thesis, M. Kamgarpour covered the topic of Air Traffic Control challenges, and the proposal of incorporating Trajectory Based Operations (TBO) into future Air Traffic Management (ATM) infrastructure [13]. Such proposals are currently already addressed in



Europe within the framework of the Single European Sky ATM Research (SESAR) [1] and also in the United States with NextGen. [14]. In TBO, individual flights would have the freedom to adjust their trajectories according to real time traffic and weather conditions. The research by Kamgarpour proposes a hybrid system modeling framework, and the task of optimal trajectory design is formulated as a hybrid optimal control problem subject to constraints of collision and hazardous weather avoidance. Additionally, a case is studied where an optimal runway scheduling for a major airport subject to weather constraints is posed as a hybrid optimal control problem, where in the hybrid system the runway configuration represents the discrete modes and the current airspace situation represents the continuous state.

### **2.3.2 Hybrid Optimal Control Approach to Commercial Aircraft Trajectory Planning (M. Soler, 2010) [15]**

Several papers have also been published by M. Soler investigating feasible use cases for using a hybrid optimal control approach to solve relevant optimization problems in aviation. In [15], an application for commercial aircraft trajectory planning is presented. It is recognized that a generally suboptimal flight profile is being flown by commercial aircraft due to the need to fit aircraft trajectories to the ATM system requirements. By incorporating different flight phases and optimal procedures into a single optimal control problem using HOC theory, it is aimed to improve the efficiency of flight trajectories.

### **2.3.3 Multiphase Mixed-Integer Optimal Control Approach to Aircraft Trajectory Optimization (Bonami et al, 2013) [16]**

P. Bonami released a paper in 2012 which discusses and introduces a methodology for designing and solving a Hybrid Flight Trajectory optimization problem with both discrete and continuous variables. The studies presented in this paper are of major importance since several concepts discussed in earlier studies ([17][15]) are combined to arrive at a method which can find an optimal solution for a more general level of HOC problems. A fifth degree Gaus-Lobatto collocation method is used to discretize the continuous problem. The discrete part of the problem is dealt with in two stages; first, the unknown switching times are dealt with by making them part of the state, and introducing a new independent variable with respect to which the switching times are fixed. This is the same approach as introduced by M. Soler [15]. The resulting MINLP problem, where the step sequence is yet to be determined, is then dealt with using a branch-and-bound algorithm as discussed in Section 2.4.

### **2.3.4 Flight trajectory design in the presence of contrails: Application of a multiphase mixed-integer optimal control approach (M. Soler et al., 2014) [18]**

In another paper by M. Soler et al. [18], further research is presented which involves using the already established method for solving HOC problems introduced by P. Bonami discussed in Section 2.3.3 [16]. In this case, the discussed application is contrail formation, which is identified as a possible cause for the greenhouse effect due to the reflective properties of the contrail clouds. The discrete aspect in this optimization problem is the flight level at which the aircraft flies, which can change in steps of 200 ft. between FL270 and FL410 when it is beneficial with respect to minimizing contrail formation. An interesting extension to previous work is discussed here, since this HOC problem involves an unknown sequence of switches of flight level. Therefore, a branch-and-bound method is implemented to find the

most optimal combination of discrete switches and their timings. This branch-and-bound method is based on work presented by P. Bonami [16], [19], [20], to which Soler has also contributed.

### **2.3.5 A Hybrid Optimal Control Approach to Fuel-Efficient Aircraft Conflict Avoidance (M. Soler, M. Kamgarpour et al., 2016) [21]**

Soler and Kamgarpour also collaborated to formulate a fuel-optimal conflict-free aircraft trajectory planning as a hybrid optimal control problem, where the discrete aspect of the system represents the air traffic procedures for conflict resolution, e.g. speed and turn advisories. The purpose of such a method is that it allows for TBO in ATM to be more feasible, since an optimal conflict resolution is generated by the algorithm and then available to all aircraft for execution. This research can be seen as a follow-up to the work presented in [22].

An interesting aspect of this research application of hybrid optimal control is that a multi-agent case study is considered here, with multiple aircraft being incorporated into the hybrid system to compute trajectories for each aircraft which together result in an overall optimal performance. To deal with this increased number of aircraft and therefore the number of binary variables in the system, it was decided to not use the branch-and-bound techniques as implemented by M. Soler in his previous research [18]. Branch-and-bound was deemed unsuitable since the MINLP becomes intractable with a large number of binary variables. Also, for branch-and-bound, the number of switches must be set *a priori*, but in this case study this would be impractical. Instead, the approach was taken to relax the binary functions associated to the mode switches from a discrete  $\{0,1\}$  domain to a continuous  $[0,1]$  domain, effectively turning the problem into a standard constrained optimal control problem. To ensure that only trajectories are taken which do satisfy the original binary constraints (and are therefore feasible in real life), a penalty weight is introduced to solutions that are non-binary. Through this approach, *off-the-shelf* NLP solvers can be used.

## **2.4 Analysis of currently common HOC problem-solving techniques**

In the previous sections, the principles of hybrid optimal control and its applications in aviation trajectory optimization have been discussed. In this section, an analysis and comparison is presented of the methods for dealing with the discrete part in hybrid optimal control problems that have proven to be successful and practical. This way, a trade-off matrix can be made between the methods, such that it can easily be determined which method is appropriate for which problem, and which method should be considered for this thesis.

### **2.4.1 Branch-and-bound**

One of the most common methods in use for discrete and combinatorial optimization problems is the so-called branch-and-bound paradigm. branch-and-bound has proved itself for decades in Mixed Integer Linear Programming (MILP)[19]. Using a branch-and-bound approach to solve fairly general hybrid optimal control problems was proposed by O. von Stryk and M. Glocker [2] in 2000, and several researchers have since demonstrated how branch-

and-bound can be applied to optimize the discrete aspects of a hybrid optimal control problem present in their own work such as [16] and [18], which were already discussed in Section 2.3.

Branch-and-bound is a decomposition approach which aims to reduce the number of iterations needed when calculating an optimal control. This is achieved by solving sequences of problems providing rigorous upper and lower bounds which are applied adequately to each problem in the sequence. This way, an explicit enumeration of the entire feasible discrete control space is avoided, saving significant computing time [17]. Sometimes, the feasible discrete control space is of too large size, making this ‘naïve’ approach of just calculating the outcome of every combination impossible.

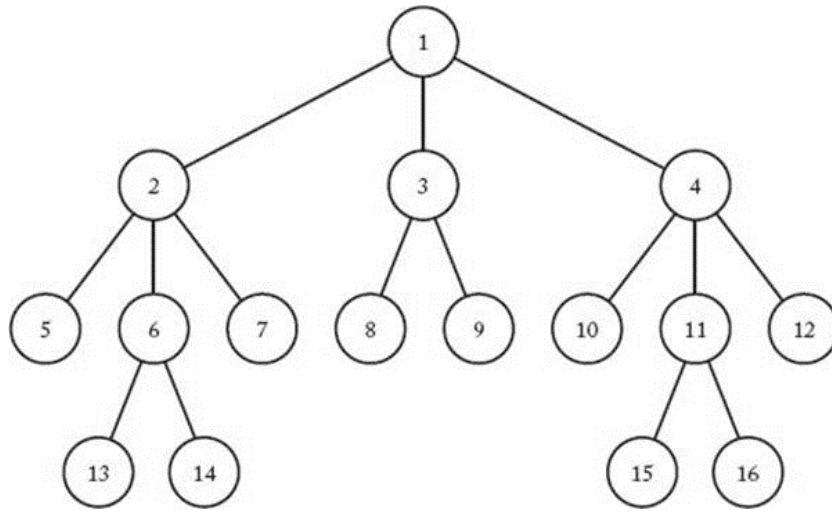


Figure 6: Example of a branch-and-bound search tree.

A requirement for the method is that a solution to relaxed versions of this hybrid optimal control problem (the RIOCP) – where (most of) the discrete switches relaxed to the continuous domain – exists. These solutions provide the lower bounds  $\beta_L$  to the branch-and-bound discrete tree  $\mathcal{T}$ . Step by step, this continuous solution is then narrowed down to satisfy the discrete constraints of the HOC by making one more node  $\mathcal{N}$  in the search tree discrete. Then, the resulting relaxed problems are solved for again. An upper bound  $\beta_U$  is also introduced, initially with value  $+\infty$ , and this upper bound is updated every time a new more optimal integer-feasible solution is found. In the end, the algorithm should have found a solution  $v^*$  along a ‘tree branch’ which is the most optimal while satisfying all integer constraints. Pseudo-code for NLP-based branch-and-bound as presented by Bonami et al. [16] [19] is presented in Algorithm 1:

---

---

**0. Initialize.**

$\mathcal{T} \leftarrow \{(\emptyset, \emptyset)\}$ .  $\beta_{\mathcal{U}} = \infty$ .  $v^* \leftarrow \text{NONE}$ .

**1. Terminate?**

Is  $\mathcal{T} = \emptyset$ ? If so, stop and return the sequence described by  $v^*$ .

**2. Select.**

Choose and delete a problem  $\mathcal{N}^l = (\mathcal{L}^l, \mathcal{U}^l)$  from  $\mathcal{T}$ .

**3. Evaluate.**

Solve the RIOCP  $(\mathcal{L}^l, \mathcal{U}^l)$ . If no solution can be found, go to step 1, else let  $\beta_{\mathcal{L}^l, \mathcal{U}^l}^R$  be its objective function value and  $\hat{v}$  be the values for the relaxed binary variables.

**4. Prune.**

If  $\beta_{\mathcal{L}^l, \mathcal{U}^l}^R \geq \beta_{\mathcal{U}}$ , go to step 1. If  $\hat{v} \in \{0, 1\}^{N-1} \times \{0, 1\}^{n_{v^g}}$  go to step 5, else let  $\beta_{\mathcal{U}} \leftarrow \beta_{\mathcal{L}^l, \mathcal{U}^l}^R$ ,  $v^* \leftarrow \hat{v}$ , and delete from  $\mathcal{T}$  all problems with  $\beta_{\mathcal{L}}^k \geq \beta_{\mathcal{U}}$ . Go to step 1.

**5. Divide.**

Create two new nodes  $\mathcal{N}^{|\mathcal{T}|}$ , and  $\mathcal{N}^{|\mathcal{T}|+1}$ . Choose  $\hat{q}$  and  $\hat{j}$  such that

$\hat{v}^{\hat{q}, \hat{j}} \in \{0, 1\}$ . Let  $\beta^{|\mathcal{T}|} \leftarrow \beta^{|\mathcal{T}|+1} \leftarrow \beta_{\mathcal{L}^l, \mathcal{U}^l}^R$  and add the problem

$\mathcal{N}^{|\mathcal{T}|} = (\mathcal{L}^l \cup v^{\hat{q}, \hat{j}}, \mathcal{U}^l)$  and  $\mathcal{N}^{|\mathcal{T}|+1} = (\mathcal{L}^l, \mathcal{U}^l \cup v^{\hat{q}, \hat{j}})$ . Go to step 1.

---

---

Algorithm 1: NLP-Branch-and-bound pseudo-code [16].

A major complication to branch-and-bound is the fact that finding suitable lower bounds through solving the relaxed problems is not straightforward, because the relaxed problems are often nonconvex, whereas a convex approximation of the optimization problem is required. [16] has experimented with systematic methods which compute convex approximations for nonlinear systems, but without much success. Instead, a heuristic approach with approximate solutions for the lower bounds was applied. It was admitted that because of this, the final solution is also not exactly optimal, and that the quality of the approximation is highly important for the quality of the final solution.

It can however still be said that branch-and-bound is a promising means of extending the solving capabilities of earlier discussed direct collocation methods used for multiphase problems, by allowing not only the discrete switching times to be unknown but also the sequence of the switches (provided that a good initial guess for the upper and lower bounds is available). This makes this method very appealing, since it could be built on top of existing numerical programs and thus not everything needs to be built from the ground up.

### 2.4.2 Relaxation of binary constraints through a penalty term

This approach was applied in [21] to deal with hybrid problems in a simpler way than branch-and-bound, because the HOC problem is essentially converted into an ordinary OC problem. This procedure is as follows:

The hybrid optimal control problem is formulated as a mixed integer optimal control problem of the following form:

$$\min J(u, w) = \Phi(x(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t), w(t)) dt \quad (2.22)$$

subject to:

$$x(t_0) = x_0 \text{ and } \forall t \in [t_0, t_f] \quad (2.23)$$

$$\dot{x}(t) = \bar{f}(x(t), u(t), w(t)) \quad (2.24)$$

$$\bar{h}(x(t), u(t), w(t)) \leq 0 \quad (2.25)$$

$$w_q(t) \in \{0, 1\}, q = 1, \dots, n_q \quad (2.26)$$

$$\sum_{q=1}^{n_q} w_q(t) = 1 \quad (2.27)$$

Instead of discretizing the problem at this stage to form a MINLP, the binary constraints are first relaxed, similar to the approach used in branch-and-bound techniques to find lower bounds, thus,  $w_q(t) \in [0, 1]$ . Then, to incorporate the penalty for non-binary results, auxiliary optimization variables are introduced for every relaxed constraint as  $\beta_q : [t_0, t_f] \rightarrow [-1, 1]$  for  $q = 1, \dots, n_q$  with  $\beta_q(t) = 2w_q(t) - 1$  being the relationship between  $\beta_q$  and  $w_q$ . Next, a penalty cost function  $l : [0, 1] \rightarrow \mathbb{R}_{\geq 0}$  is defined as a strictly monotonically decreasing function with  $l(1) = 0$ . Then, the MIOCP can be rewritten as a classical optimal control problem as:

$$\min J(u, w) = \Phi(x(t_f)) + \int_{t_0}^{t_f} \left( L(x(t), u(t), w(t)) + \alpha \sum_{q=1}^{n_q} l(|\beta_q(t)|) \right) dt \quad (2.28)$$

subject to:

$$x(t_0) = x_0 \text{ and } \forall t \in [t_0, t_f] \quad (2.29)$$

$$\dot{x}(t) = \bar{f}(x(t), u(t), w(t)) \quad (2.30)$$

$$\bar{h}(x(t), u(t), w(t)) \leq 0 \quad (2.31)$$

$$\beta_q(t) \in [0,1], q = 1, \dots, n_q \quad (2.32)$$

$$w_q(t) = \frac{1}{2}(1 - \beta_q(t)), q = 1, \dots, n_q \quad (2.33)$$

$$\sum_{q=1}^{n_q} w_q(t) = 1 \quad (2.34)$$

This way, no discrete elements remain present and *off-the-shelf* NLP solvers can be used to discretize the OCP and solve it. By choosing  $\alpha$  sufficiently large (approaching infinity), all (infeasible) non-binary controls for  $w$  are eliminated from the solution space.

An important advantage over branch-and-bound is the relatively low complexity of implementing it, because only the problem has to be rewritten. However, rewriting the problem could be an impractical approach in light of this thesis research. In general, this method is relatively new in terms of being applied to aircraft trajectory optimization problems and thus less proven, but very promising.

### 2.4.3 Conversion of integer constraints to decision variables

Another method of tackling the optimization of hybrid (switched) systems is the parametrization of discrete constraints. This approach was first introduced by X. Xu and P.J. Antsaklis [23], and was later applied to an aircraft trajectory optimization problem with unknown switching times for the various flight phases by M. Soler et al. [15] as discussed in Section 2.3.2.

For a switched system where the sequence of the switches is known and the number of switches is  $N$ , but where the switching times are to be determined, the process of applying this technique is as follows (based on [15]):

1. Introduce new state variables  $x_{n+1}, \dots, x_{n+N}$  for each switching moment to the state vector  $x = [x_1, \dots, x_n]$ . The new extended state vector will then be  $\hat{x} = [x_1, \dots, x_n, x_{n+1}, \dots, x_{n+N}]$ . In the end, values for these added state variables will be obtained by the optimization model and these will represent the optimal switching times  $t_i$ .

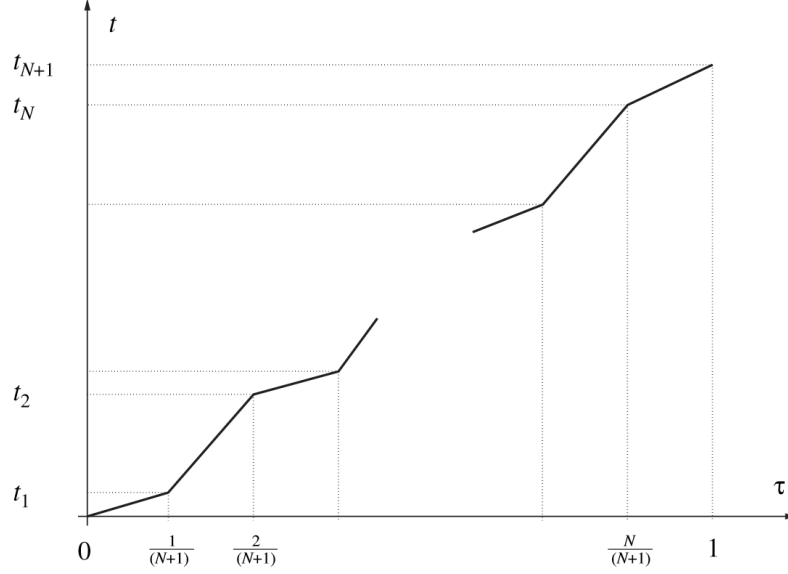


Figure 7: Relation between scaled time  $\tau$  and real (unscaled) time  $t$  [15]

2. Introduce the new independent variable  $\tau$ , which will replace  $t$  in the original problem and has a value in  $[0,1]$ . The relation between  $\tau$  and  $t$  will be different on each interval between switching moments, but will always be linear. A valid expression for the relation between  $t$  and  $\tau$  is as follows:

$$t = \begin{cases} (N+1)x_{n+1}\tau & 0 \leq \tau \leq 1/(N+1) \\ (N+1)(x_{n+i+1} - x_{n+i})\tau + (i+1)x_{n+i} - ix_{n+i+1} & i(N+1) \leq \tau \leq (i+1)/(N+1) \\ (N+1)(x_{n+N})\tau + (N+1)x_{n+N} - N & 1/(N+1) \leq \tau \leq 1 \end{cases} \quad (2.35)$$

Where  $i = 1, \dots, N$ .

3. Then, the cost function can be written as

$$J = \phi(\hat{x}(1)) + \int_0^{\frac{1}{N+1}} \widehat{L}_1(\hat{x}, u, \tau) d\tau + \dots + \int_{\frac{N}{N+1}}^1 \widehat{L}_N(\hat{x}, u, \tau) d\tau \quad (2.36)$$

Where

$$\widehat{L}_i(\hat{x}, u, \tau) = (N+1)(x_{n+i+1} - x_{n+i})L(x, u, t(\tau))$$

On each interval  $i(N+1) \leq \tau \leq (i+1)/(N+1)$ .

The problem of minimizing  $J$  is now a conventional control problem subject to the parametrized system

$$x' = (N+1)(x_{n+i+1} - x_{n+i})\widehat{f}_i(x, u, \tau) \quad (2.37)$$

, where  $\widehat{f}_i(x, u, \tau) = f_i(x, u, t(\tau))$ , and also subject to any corresponding path constraints of equality or inequality form. The resulting optimal trajectory  $\hat{x}^*$  will then also describe the optimal switching times in the last  $N$  components.

This is thus also a method in which instead of trying to solve HOC problems in their original form, one manipulates the problem into a form which can be solved by regular optimal control solving methods. An issue with this methodology is that when there is more than one discrete variable present in the system (i.e. also an unknown switching sequence), the problem becomes extremely complex and the computational effort required will likely be too much. However, as demonstrated in [16], using this technique for the switching times at least reduces the complexity of the discrete aspect of the HOC by one dimension, allowing for a less rigorous implementation of additional methods which deal with the remaining discrete variables (such as branch-and-bound). Given that GPOPS is already capable of solving this degree of hybrid problems by linking phases but leaving the switching times free, implementing this approach is not necessary.

#### **2.4.4 Assuming the switching structure**

This approach is how multiphase optimal control problems are currently already being dealt with in optimal control software such as GPOPS. Instead of determining the switching structure through an algorithm, linkage constraints are defined for the switching moments. Depending on the problem, it might be more practical to use this approach when one already has a good idea of the most optimal switching structure [2]. Also, assuming the switching structure can be used as a first step in HOC problems, as a means of obtaining an initial guess for the ultimately optimal solution.

#### **2.4.5 Comparison matrix of HOC solving techniques**

Based on the studied literature concerning each of the techniques described in this chapter, one can make a qualitative assessment of the usefulness of each of them when analyzing them with respect to certain criteria. This assessment is presented in Table 1. It should be noted that there is no clear ‘winner’, with each method having its potential strengths and drawbacks. Also, an important remark is that these methods do not exclude each other, as they could be used in combination. However, with regards to this thesis research, branch-and-bound is the most promising method since it can be implemented without modifying the problem significantly, while allowing for a much greater degree of hybrid problem to be solved.



Table 1: Assessment of HOC solving techniques

	Branch-and-bound	Relaxed integers to conventional OCP	Convert constraints to decision variables	Assume switching structure and solve regularly
Applicability	+/-	+/-	-	++
Accuracy	++	+	++	--
Reliability/ convergence	+/-	+	+	+
Computational efficiency	+	++	++	++
Complexity	+	+	++	+
Ease of implementation	-	+/-	+	+



## 3 Formation flight and its application for civil aviation

Over the last decades, a growing amount of interest has been shown in researching the aerodynamic benefits of formation flying. Based on the observation made by Lissaman et al. [7] that birds who were flying together in a V-shaped formation could achieve a range benefit of as much as 70%, theories were formulated which described the background of this aerodynamic advantage. These theories were later also applied to aircraft.

### 3.1 Aerodynamic concept

The aerodynamic benefit of formation flight is obtained as a result of wing tip vortices generated by the leading aircraft (Figure 8). These vortices have an upward vertical component (*upwash*) on the outboard side with respect to the leading aircraft as well as a downward vertical component (*downwash*) on the inboard side (Figure 9). A trailing aircraft benefits from this wing tip vortex generated ahead by flying through the upwash region. This causes the velocity vector of the air relative to the trailing aircraft to have an upward component, resulting in a forward component of the generated lift and an upward component for the generated drag (with inherently a smaller rearward component) (see Figure 10). As a result, less engine thrust is required for the trailing aircraft to maintain steady flight conditions, such that fuel can be saved. Essentially, the trailing aircraft is ‘surfing’ the upwash of the wingtip vortices of the leading aircraft. This aerodynamic effect results in a reduction of induced drag for the trailing aircraft.

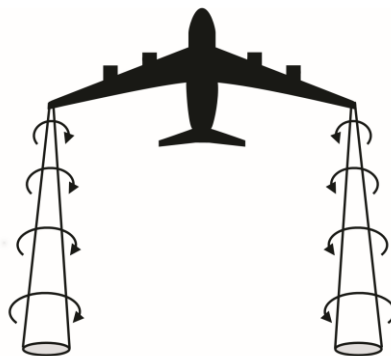


Figure 8: Visualization of wing tip vortices generated by an aircraft.

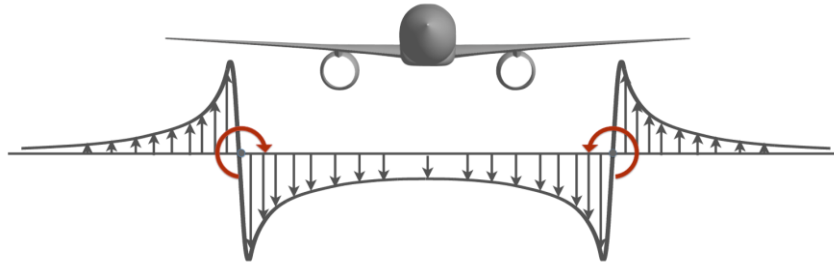


Figure 9: Vertical component of air velocity induced by an aircraft wingtip vortices behind an aircraft [24].

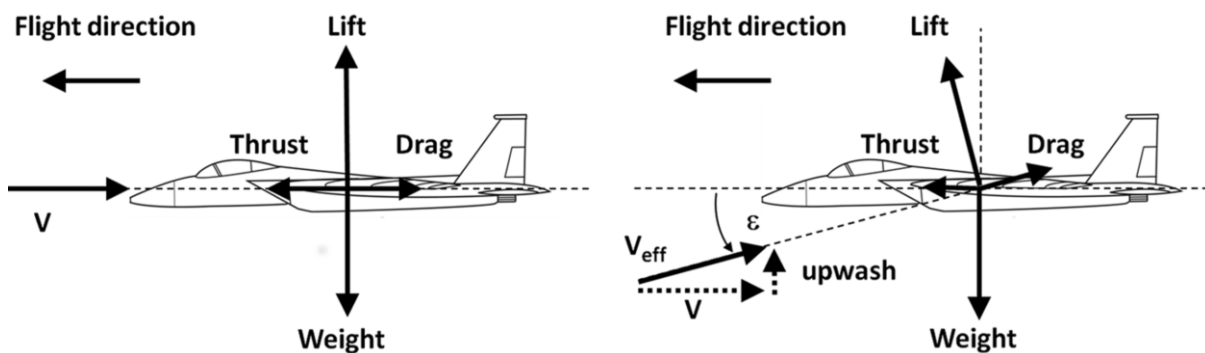


Figure 10: Effect of upwash on thrust required for forward flight [25].

Another effect of flying through the wake of the leading aircraft is that there is an asymmetric load on the trailing aircraft's wings, since the upwash effect is smaller at the outward wing relative to the formation. This in turn creates a roll effect, which needs to be compensated for by the trailing aircraft through trimming. The increased drag generated by extra control surface usage is an aspect which needs consideration when determining the total aerodynamic benefit.

### 3.2 Positioning between aircraft in formation

Blake et al. researched how a formation of aircraft which are positioned within a few wingspans from each other (close formation) could lead to savings in induced drag [26]. It was concluded that significant reductions could be achieved in an optimal formation structure. Because having multiple aircraft fly close to each other within a formation would clearly violate current safety regulations due to a high risk of mid-air collisions, it is of importance to investigate whether the positive aerodynamic effect of formation flying can also be obtained with a safe streamwise separation between aircraft. Given the fact that many aircraft in cruise conditions generate wakes which persist for long distances, S.A. Ning et al found that in atmospheric conditions with low to moderate turbulence levels, a streamwise separation of up to 40 wingspans could still result in significant reductions in induced drag levels [27]. At a 30 wingspan streamwise separation distance, a maximum of  $40 \pm 6\%$  total formation induced drag reduction was estimated to be achievable in a three-aircraft formation. With this

perspective, civil aviation formation flying became a much more feasible approach to reducing fuel consumption.

Laterally, a high degree of variance in aerodynamic benefits is observed. An accurate relative lateral and vertical position of the trailing aircraft with respect to the leading aircraft is considered highly important for a successful formation flying operation [26]. Optimally, the lateral spacing between the aircraft wing tips is slightly negative [27]. However, if it is too negative, a part of the trailing aircraft wing will experience a negative downwash effect, so this should be avoided.

With this knowledge, it is of interest to determine the optimal formation shape for a multi-aircraft formation. [24] concluded that a V-formation is the most robust in extended formation flight, with the highest induced drag savings even in less optimal conditions. However, the inverted-V formation shape, although more sensitive to positioning errors, has some beneficial characteristics such as a more symmetric loading for the trailing aircraft, resulting in less roll behavior. Of course, V or inverted-V formations are only possible with a formation of three aircraft or more. In a two-aircraft formation, the echelon formation is the best setup.

In this thesis, the exact shape of the formation nor the relative spacing between the aircraft are a part of the model. However, numerical values which are used are derived from the characteristics of an extended V-shape formation flight described here.

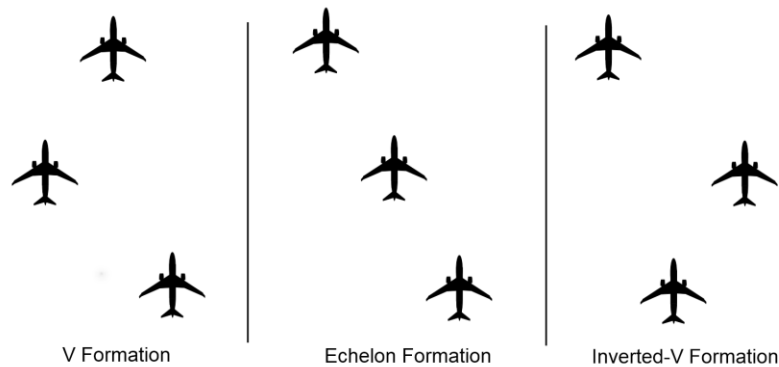


Figure 11: Three common formation shapes [27].



## 4 Optimization problem description

With the obtained knowledge about optimal control, hybrid optimal control and formation flight, a trajectory optimization problem can be formulated which deals with formation flight using hybrid optimal control techniques. This chapter describes the problem and assesses its validity as an application of hybrid optimal control theory.

### 4.1 Problem background

When it comes to the implementation of formation flight in real world civil aviation operations, a planning problem arises concerning the optimal approach to schedule and execute formation flights with the highest economic and environmental benefit. Given the sheer number of candidate flights which are performed every day between, for example, Europe and North-America, there is a need to find a method which selects the sets of flights which benefit the most from joining each other in a formation. Additionally, the problem of finding the best trajectory for all aircraft joining this formation, including their rendezvous and split-up locations, to generate the highest cumulative fuel savings is also introduced. These two aspects combined create a highly complex problem, where generally a trade-off is made between problem size (i.e. number of flights considered) and the fidelity of the trajectory.

In previous research, this trade-off has been made in various ways. Van Hellenberg Hubar, Hartjes et al. [3] developed a trajectory optimization tool which optimizes the 4D trajectories for two aircraft joining in an extended formation for the minimum overall fuel consumption with a relatively high fidelity. The aircraft dynamics are modelled with a reduced-order point mass system. Case studies performed with this tool gave satisfying results and yielded significant fuel consumption reductions with a limited increase in flight time. However, it was noted that having only two aircraft modelled was a limiting factor. It was recommended that larger formations were investigated, but in the current framework it would prove too time consuming to run such multi-aircraft simulations.

H.G. Visser et al. [28] performed a simulation with a very large number of flights being modeled as agents which could, if the conditions at that moment are beneficial, join a formation en-route with aircraft flying a similar trajectory nearby. This decentralized approach is especially useful when a high degree of flexibility is required: formation flying is in this case only seen as an option rather than an objective. However, major simplifications were made to the flight model: A constant altitude and velocity are assumed, so the flights are modeled as being permanently in cruise mode. Moreover, the fuel usage benefit was expressed as a constant 10% reduction for trailing aircraft, rather than a value as a function of the induced drag reduction. Also, it was concluded that that suboptimal formations were formed as agents were modeled to always join the first viable formation.

## 4.2 Problem statement

The research presented in this thesis, as mentioned in the research goals, aims to find a new optimization framework which is positioned between the research performed in [3] and [28]. This is to be achieved by designing it such that a simulation with four aircraft can be performed, enabling four-aircraft formations as a possibility. To achieve accurate trajectories, all aircraft are to be modelled with a reduced order point mass system.

Although four aircraft is mainly chosen as the maximum because of the dramatic increase in complexity introduced with a problem of five or more aircraft, it has been mentioned in [27] that in formations of a size larger than four, the additional aircraft will have relatively less benefit from being in the formation. In [28], this reduced benefit for large formations is not taken into account, which resulted in many formations of size seven and larger. If this was modelled more accurately, such large formations would possibly not have occurred. Also, from a practical standpoint, the feasibility of coordinating and safely operating formation flight sizes larger than four is questionable.

Compared to [28], a higher degree of accuracy can be achieved, since the aircraft dynamics are to be modelled as a point mass system. However, the flexibility of a decentralized approach will be lost with this centralized method, which is based on pre-planning. Since one of the conclusions of [28] is that even with their decentralized model, a certain degree of pre-planning would be beneficial for finding more optimal formation pairs, this disadvantage is readily accepted. With a simulation size of only four aircraft, a methodology will have to be developed separately which finds the four best candidate flights to be modelled. This is however deemed beyond the scope of this thesis.

## 4.3 Optimization goal

In an optimization problem, it is regularly the case that a single optimization objective is not clearly definable. In the case of formation flying, there are several aspects in addition to fuel burn which could influence the definition of an optimal trajectory. First, aircraft will deviate from their shortest great circle route to join the formation. This, together with the fact that cruise speeds in formation are generally slightly lower than in solo flight, means that the flight time will increase. This can be seen as a disadvantage for travelers, and could pose a planning problem for airlines as well, with potentially higher operating costs. Additionally, the fuel saving benefits are not distributed evenly among the participating aircraft in the formation. The leading aircraft of the formation will experience an increase in fuel burn due to the extra distance flown to accommodate the formation, while the aircraft flying in the most rearward position will profit the most. If only individual interests of aircraft are considered, this would mean that the likelihood that an aircraft is prepared to lead a formation is rather minimal. Lastly, the robustness of the formation trajectory is another factor to consider, since if a disturbance in one of the participating flights or a small change in one of the input parameters would heavily influence the formation feasibility, the solution outcome might not be reliable.



In this optimization problem, it is chosen to simplify the optimization goal to a minimization of only the cumulative fuel usage of all aircraft over their entire respective flights. In a cost function, this is expressed with a Mayer term as the difference between all cumulative final aircraft weights and their respective initial weights:

$$\text{minimize } J = \Phi[x(t_0), t_0, x(t_f), t_f] \quad (4.1)$$

where:

$$\Phi = \sum_{i=1}^n (W_i(t_0) - W_i(t_f)) \quad (4.2)$$

for  $n$  number of simulated flights.  $W$  is a part of the aircraft state  $x$ .

However, the mentioned additional considerations can still be kept in mind when analyses are made about the obtained results. For instance, making a qualitative assessment about the total extra flight time compared to the solo flight scenario can be valuable for validating the usefulness of the results. Similarly, a sensitivity analysis can be performed to determine the robustness of the found solution.

#### 4.4 Application of hybrid optimal control theory

The problem statement in Section 4.2 poses a challenge in managing the complexity of the problem, since the number of possible solutions increases dramatically with additional aircraft. It is for this reason that the presented problem statement provides an excellent opportunity to approach the problem as a hybrid problem. Discrete switches occur in the trajectory each time when an aircraft joins or leaves a formation, and the timing, the sequence and the number of the switches are to be determined. This exactly matches the definition of the hybrid optimal control problem presented in Section 2.2. Finding a suitable method which deals with this combinatorically complex problem in an efficient manner is the main challenge of this thesis project. With the cost functional already defined in Section 4.3, the hybrid optimal control problem considered in this thesis otherwise follows the format of the hybrid optimal control problem presented in Section 2.2.1.



## 5 Methodology for trajectory optimization

Now that the optimization goal and the hybrid optimal control problem clearly defined, a numerical implementation is to be developed. This chapter describes the groundwork which was used to build the optimization tool, as well as the chosen approach with respect to the hybrid optimal control functionality of the developed tool.

### 5.1 Optimization tool requirements

Based on the research goals stated in Section 1.4 and the optimization problem description of Chapter 4, requirements can be formulated for the hybrid optimal control trajectory optimization tool. These requirements are presented below:

1. The tool should work with existing optimal control software.
2. The tool should enable formation flight trajectory optimization for three or more aircraft.
3. Computational times must remain within a practical limit.
4. Model should generate an accurate approximation of a real-world trajectory using realistic flight dynamics.
5. A smart way to reduce the combinatorial complexity of HOC problems should be implemented.
6. The tool should be built with flexibility in mind for different experiments.
7. The tool should work automatically.
8. The tool should have a clear structure which is comprehensible for future users and adaptable for future modification.

### 5.2 Existing optimal control software as a starting point

Several approaches exist towards dealing with optimization problems, however in this research a major focus lies on the approach using *optimal control theory*. There are multiple software packages available which already integrate one or more methods to solve optimal control problems, and using one of these software packages as a foundation for further research is essential to keep the size of the research project within limits. Having a user interface readily available to solve problems makes for a much easier process of testing multiple optimization cases.

In [8], several of these software packages are compared and their respective advantages and disadvantages are presented. The considered packages are *EzOpt*, *Sparse Optimal Control Software (SOCS)*, *DIDO* and *General Pseudospectral OPTimal control Software (GPOPS)*. It was concluded that for the discussed research in this publication, GPOPS is the most suitable for trajectory optimization research at TU Delft as it offers multiple options for both collocation and differentiation. Also, GPOPS is integrated in the versatile MATLAB environment, making it especially easy to use. Since the formation flight optimization problem

is in a similar field, with similar constraints in terms of operations compared to the ECHO project discussed in [8], it is concluded that also for this thesis project GPOPS is a suitable choice.

GPOPS is already capable of solving multiple-phase optimal control problems, where phases are linked as defined by the user such that the total outcome is valid for the user's mission requirements. Since GPOPS is already capable of solving multi-phase optimal control problems, where the switching times between phases can be left open to be solved for, this software package already incorporates hybrid functionality to a certain degree. In its current state, GPOPS is however not able to solve problems where the sequence and number of phases is also left open. As discussed in Chapter 1, the aim of this research project is to incorporate support for a degree of combinatorial complexity into the functionality of GPOPS, such that an overall more optimal solution to hybrid problems can be obtained.

GPOPS [4] essentially acts as an interface which translates an infinitely-dimensional optimal control problem into a discretized, finite-dimensional nonlinear programming problem (NLP) which can subsequently be solved using a NLP solver. For the NLP solver to function, the derivatives of the objective function gradient have to be supplied to the NLP solver. Several NLP solvers as well as automatic differentiation tools can be used in combination with GPOPS. In previous research [8][29][3], the NLP solver SNOPT (Sparse Nonlinear OPTimizer) has been successfully exploited in combination with the automatic differentiation software INTLAB (Interval Laboratory) to solve the designed aircraft trajectory optimization problems. This combination of solver and differentiation software is also used for this thesis. Lastly, the choice of collocation method is to be defined. Based on the studies presented in Section 2.1, the Radau Pseudospectral Method (RPM) [30] is chosen, which is also supported by default in GPOPS.

Specifically for aircraft trajectory optimization, Dr. Ir. S. Hartjes developed FORT, a fixed-wing aircraft optimization tool, a setup wrapper for GPOPS which includes aircraft dynamic models, atmospheric models, relevant parameter conversion tools and diagram generation tools. Also, templates are included for the cost function, the differential algebraic equations, the linkage file and the setup parameters. FORT is used in [3] for the optimization of the multiple-phase two aircraft formation flight trajectories, and is thus deemed to be a suitable basis for the hybrid multi-aircraft problem considered here.

## 5.3 FORT (GPOPS) working structure and limitations

### 5.3.1 Description of GPOPS input parameters

For GPOPS to process an optimal control problem, the following input parameters must be specified:

Table 2: Description of GPOPS input parameters

Input parameter	Description
Cost functional	The cost functional $J$ (Section 4.3), consisting of a Mayer term and/or a Lagrange term, defining the value which is to be minimized.
Differential equations	The right side of the differential equations which define the model state behavior as a result of the defined controls. The path of the trajectory is also defined here if constraints need to be imposed on it.
Linkage setup	Parameter which defines the linkage of the state between phases in a multi-phase problem: Which phases are linked and how are the states connected.
Linkage constraints	The state difference margins between the linked phases.
Intervals	The number of intervals at which the solution is evaluated per phase
Nodes	The number of nodes per interval. This determines the order of polynomial used to fit the collocation points.
Time constraints	Time range in which the solution can exist.
Duration constraints	Minimum and maximum durations of a phase.
State constraints (limits)	Minimum and maximum state values during a phase.
Boundary conditions	Minimum and maximum values for the state at the beginning and end of each phase and overall.
Control constraints	Limits for the controls per phase.
Path constraints	Minimum and maximum values for the path.
Initial guess	A structure in the same format as the required solution, with a guess of the expected result for the time, state, control and static parameters.
Solver setup specifications	The choice for a solver, differentiator and their respective setup parameters.

### 5.3.2 FORT working structure

The GPOPS shell FORT, specifically designed for fixed-wing aircraft trajectory optimization, incorporates a range of files which describe all the required inputs for GPOPS in a structured way which is logical for this application. Figure 12 describes how FORT processes all required inputs and uses GPOPS to obtain a solution:

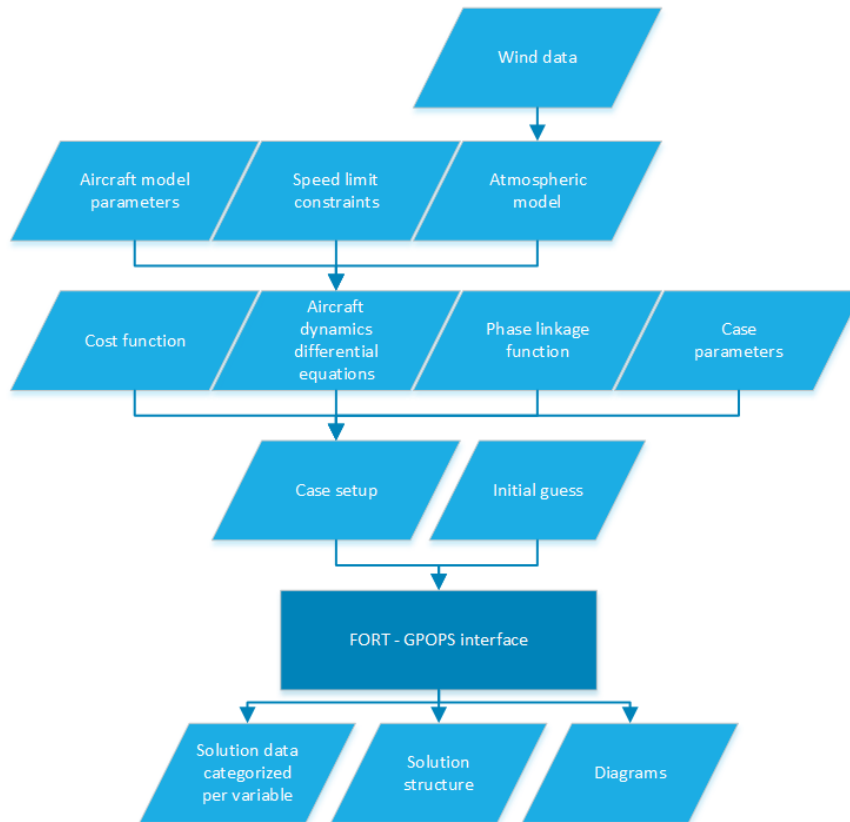


Figure 12: Working structure of FORT

### 5.3.3 Limitations of the existing FORT tool

FORT was used by [3] to perform the two aircraft multi-phase transatlantic formation flight optimization cases, and this existing set of input files were used as a starting point for this thesis. Although these setup files were designed well for a two-aircraft case with a fixed number of phases, there is no support for a flexible number of aircraft and phases. Regardless of the approach to solving the hybrid optimal control problem described in Chapter 4, FORT needs a major redevelopment to allow for a flexible case setup with just the required phase structure and the number of aircraft considered as an input.

## 5.4 Hybrid Optimal Control approach – Branch-and-bound

The end goal of this research is to successfully implement *hybrid optimal control theory*, which aims to deal with systems that are both *discrete time and continuous time* from a mathematical perspective. In this research, the methodology of using hybrid optimal control theory is not just seen as a means to an end. A successful understanding and implementation of a numerical approach to hybrid optimal control is actually defined as part of the research objective, with the actual application of hybrid optimal control to a case study being a complimentary objective. This section describes the approach which was taken to hybrid optimal control for this thesis.

### 5.4.1 Redesign of existing files

To make the existing FORT design compatible with a hybrid version of the formation flying optimal control problem, where the number of phases as well as their order and linkages are not fixed but part of the optimization, it needs to be changed in such a way that the setup files adapt automatically to any valid phase structure. To achieve this, FORT is redesigned by making all outputs dependent on the given phase structure and the number of aircraft. Also, several formatting and diagram generation improvements are made to the existing code.

### 5.4.2 Notation of formation phase structures

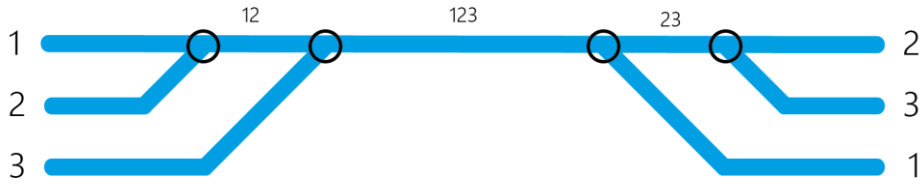
To enable a quick and flexible input for processing a range of formation phase structures, a numerical notation paradigm is developed which contains all required information for the redesigned FORT input files to generate the correct case setup. A multi-aircraft formation flight trajectory consists of five major stages:

1. Departure solo flight
2. Formation buildup: Sequentially growing formation phases
3. Formation cruise: Phase(s) containing the largest formation(s)
4. Formation breakdown: Sequentially splitting formation phases
5. Arrival solo flight

The phases of each stage are then described numerically in an array. With a given number of aircraft  $N$ , each aircraft is named with an integer number  $1, 2, \dots, N$  respectively. Then, formations are described by a concatenation of the respective aircraft names, e.g. '12' describes a formation phase with aircraft 1 and 2. Solo flight stages 1 and 5 are denoted with the value '0', indicating that no formations are present and  $N$  phases containing one aircraft are to be generated. Using this notation, a complete formation flight phase structure can be denoted in an array. One three-aircraft example looks as follows:

[     0           12           123           23           0     ]

Here, aircraft 1 and 2 join each other before aircraft 3 completes the formation (stage 2). During breakdown (stage 4), aircraft 1 leaves the formation first, so aircraft 2 and 3 remain until they eventually split up again as well. The phase structure based on this array notation is graphically represented in Figure 13. It should be noted that stage 2 and 4 will consist of multiple array elements (one for each aircraft joining or splitting moment) when more than three aircraft are considered.



○ Discrete transition between phases

Figure 13: Graphical representation of a phase structure array.

In a four-aircraft scenario, the possibility arises that in one stage, two formations of two aircraft exist. This is then denoted as a non-integer number, with the decimal values indicating the second formation. For example, '13.24' indicates a formation phase with aircraft 1 and 3 and a separate formation phase with aircraft 2 and 4. Lastly, it should be noted that for this thesis it is assumed that an aircraft can only enter and leave one formation during a flight.

### 5.4.3 Branch-and-bound algorithm

Since in this hybrid optimal control problem, all discrete phase switches are part of the optimization, the number of possible discrete solutions (phase structures) increases dramatically with every extra added aircraft. The trajectory optimization problem for one discrete solution is already computationally heavy, so it is impractical to try and run the optimization with all possible phase structures (347 for a four-aircraft case) and simply select the best-found solution. A smarter way to deal with this combinatorial problem is a branch-and-bound algorithm, which was discussed in Section 2.4.1. This method reduces the amount of evaluated phase structures by selecting only the promising ones. In Section 2.4, several other methods of dealing with hybrid optimal control problems were also analyzed, however they are deemed unsuitable for application in this thesis since they can't be modelled as an external algorithm but rather require a rewriting of the original problem.

#### 5.4.3.1 Concept

In the branch-and-bound algorithm applied to formation flying, the discrete solution space is categorized in a tree structure where each depth level of the tree represents an additional discrete variable. For formation flight phase structures, this tree is generated as follows: The top node is 'Case 0', with all flights being flown solo. Then, on the second level, intermediate phase structures are generated based on the number of aircraft  $N$ , where only stage 1, 3 and 5 are included. This means that stage 2 (buildup) and 5 (breakdown) are not considered yet and all aircraft join and leave the main formation (stage 3) simultaneously. In a three-aircraft case, this level would have four nodes; one for the three-aircraft formation, and three nodes for each possible two-aircraft formation, with the remaining aircraft flying solo completely. From the next level onwards, a formation phase is added step by step while alternating between the buildup and breakdown stage, until no more phase switches exist where more than one aircraft joins or leaves a formation at the same time. For a three-aircraft case, the solution tree which is to be evaluated is presented in Figure 14.



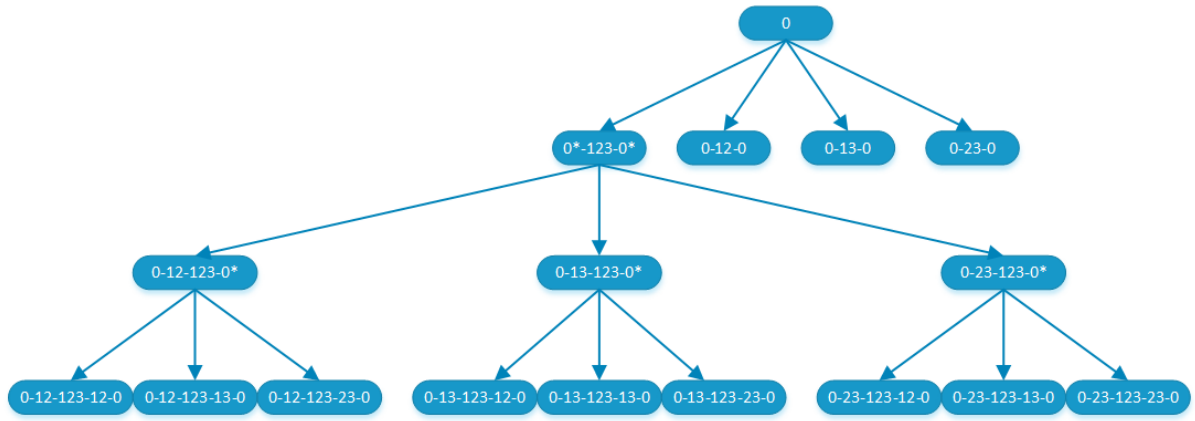


Figure 14: Solution tree for a three-aircraft formation flight case. 0\* indicates that phases are still to be added on here further down the tree.

#### 5.4.3.2 Assumptions for bounding (cut off) conditions

Generally, a requirement for the method is that a solution to relaxed versions of this hybrid optimal control problem – where (most of) the discrete switches are relaxed to the continuous domain – exists. These solutions provide the lower bounds  $\beta_L$  to the branch-and-bound solution tree. However, for formation flight it is practically impossible to model the discrete switches on a continuous domain, since there is no easily modelled physical representation of being ‘somewhere between solo and in formation’. To overcome this, a more heuristic approach can be applied (which was also done in [16]). In this case, the assumption is made that, with a defined margin, the node(s) with the best solution(s) at tree level  $i$  will also yield the most optimal solution at the bottom of the tree. This assumption is made based on the observation that:

- Per two tree levels which are evaluated, the scale at which the solution is further optimized reduces drastically
- Due to the alternation between buildup and breakdown per level, the solutions at one level (buildup) will have no significant influence on the evaluation of the next level (breakdown).

Still, this assumption will have to be validated after every run by inspecting the generated solution tree.

The algorithm is now able to chop off all the branches attached to the nodes which yield a cost functional which is higher than the best cost functional of that level plus the defined margin. Because of this, major portions of the tree can be disregarded and won’t have to be evaluated at all, which results in a significant computation time benefit.

#### 5.4.3.3 Implementation

For this thesis, the branch-and-bound algorithm needs to be implemented to work with the existing FORT environment. This is achieved by running the algorithm as an external shell

which calls FORT with the input parameters of the currently evaluated node. A flowchart for the developed Formation Flight Branch-and-bound algorithm is presented in Appendix A.

#### 5.4.4 Initial guess adaptation algorithm

Since every node in the generated solution tree contains a unique phase structure, every node also requires an initial guess in the exact format of its phase structure. It is therefore necessary to create unique initial guesses for each node. Doing this manually would be a daunting task, so an algorithm is developed which generates a valid initial guess in the format of the node which is to be evaluated, based on an original initial guess in a different format. In addition to being in the right format, it is also essential that the initial guess approximates the final solution closely enough such that the optimization tool converges quickly into the global optimum. Otherwise, the found solutions might be suboptimal (local optimum), not valid at all, or the computation time is increased drastically.

In the branch-and-bound algorithm, for every node except the tree root node ('Case 0') the solution of its parent node is taken as an input to the initial guess adaptation algorithm. To achieve a correct and accurate modified initial guess, several methods are applied to the original initial guess: First, the original initial guess is checked for formation phases which are identical to a formation phase in the target phase structure, starting with the formation cruise phase. If this phase is found to be identical, the buildup phases and finally the breakdown phases are compared next. If identical phases are found, the solution for these phases can be kept without modification. The remaining non-identical phases of the original guess are re-arranged into a single solo flight phase for each aircraft. From these solo phases, if necessary, new formation phases are constructed by taking a fraction from the relevant solo phases and combining these fractions into a formation phase. Lastly, since the trajectories of each phase will not connect properly anymore, new time and state values are calculated where necessary to eliminate phase linkage discontinuities in the modified initial guess. Appendix B presents the general functionality of the developed guess adaptation algorithm which is used for this thesis.

#### 5.4.5 Implementation of multi-threaded computing

To improve the efficiency of the evaluation of the hybrid optimal control problem even more, a method is developed to allow for multi-threaded computation using the Parallel Computing Toolbox available in MATLAB. Since it is inherent to the branch-and-bound paradigm that the number of nodes to be evaluated is not known a priori, it is initially not possible to evaluate multiple nodes independently and simultaneously using a so called *parfor loop* in MATLAB. To still be able to simultaneously evaluate multiple solution tree nodes, the evaluation is divided per tree depth level. For each tree depth level, the number of nodes will be known by the time the evaluation of that tree depth level starts. Also, since the initial guess for the evaluation of a node is by design taken from the parent node, these can also be generated independently. This way, the entire range tree depth level nodes can be evaluated using multi-threaded computation, allowing for a higher computational performance.

A drawback to this chosen approach is that parallel thread workers cannot continue to the next tree depth level when one worker is still evaluating a node in the current level. This

could cause a situation where one worker is holding up the rest of the evaluation, causing a delay in computation. Additionally, all SNOPT instances are programmed to write their solving progress log to the same file, causing a mix of log data from multiple threads. However, both these issues do not pose any major practical problems, and in this branch-and-bound algorithm, this implementation of multi-threaded computing can be used without worries.

Technically, this approach could enable the highest performance on machines with as many CPU cores as the size of the largest tree depth level. However, in this thesis only dual-core and quad-core CPUs are used.

## 5.5 Outputs and final structure

The optimization tools and algorithms described in this chapter which are developed and used, are of course designed to solve the hybrid trajectory optimization problem stated in Chapter 4. This means that the correct outputs must be generated to allow for an adequate assessment of this problem. A list of all relevant generated outputs is presented in Table 3:

Table 3: Outputs generated by the developed HOC tool

Output parameter	Description
Optimal cost functional	The minimized cost functional value $J$ (Section 4.3),
Optimal phase structure	The phase structure associated with the optimal cost functional.
Solution	A solution structure containing the time vector and, state, control and static parameters associated with the optimal cost functional.
Total calculation time	Time elapsed since starting the simulation
Tree structure with remaining nodes	A graphical and vector representation of the tree nodes which were evaluated.
Calculation times vector	Vector with the elapsed time per node calculation.
Objective value vector	Vector with the obtained objective value of each evaluated node.
Diagrams	Diagrams showing the 4-D trajectory of the optimal solution.
Solution of all nodes	Folder containing the solution structures of all evaluated nodes.
SNOPT log of all nodes	Log file of all evaluated nodes created by the solver SNOPT, containing information about the solving process.

From these generated outputs, several results of interest relevant for the problem to be solved can be derived. These are defined as follows:

- Total fuel consumption

- Fuel consumption per aircraft
- Total flight time
- Flight time per aircraft
- The performance of the optimization framework

The final hierarchical structure of the developed HOC tool, describing how several elements relate to each other in the global setup, is presented in Figure 15.

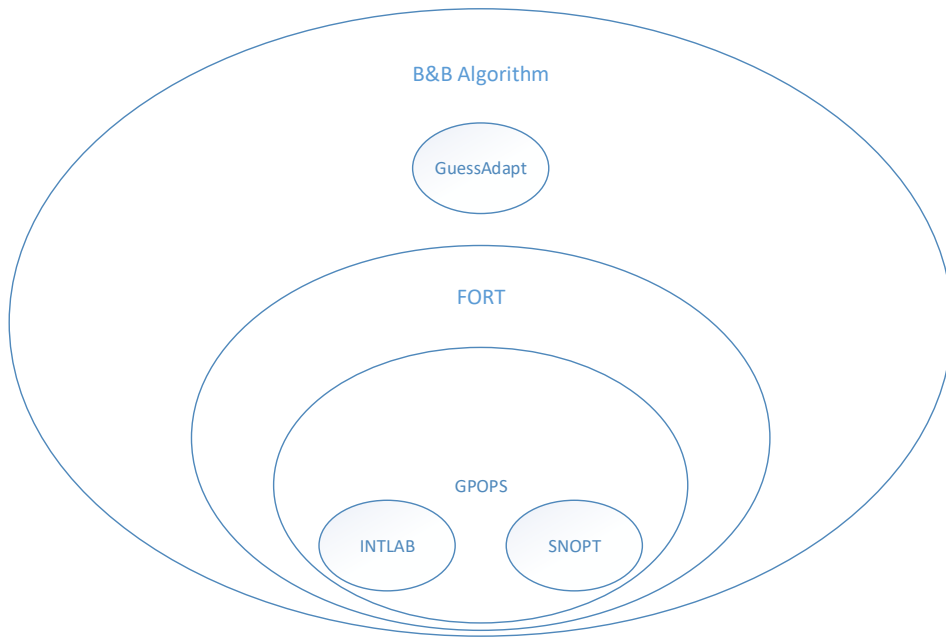


Figure 15: Hierarchical structure of various HOC tool elements

## 6 Case Studies: Experimental set-up

With the trajectory optimization framework for the hybrid optimal control formation flight problem ready, case studies can be performed in which several formation flight trajectories are evaluated. This chapter discusses the experimental set-up which is used for the performed case studies. First, it is described how the aircraft dynamics are modelled. This is followed by an elaboration on the approach which was taken to model the formation flight characteristics of the trajectory. Next, the constraints which are to be applied to the model are presented. After this, the used aircraft model parameters are discussed. It is then discussed how the cruise formation phase can be evaluated using Bréguet range equations. Lastly, an overview is given of the assumptions and simplifications which are made for the various experiments which are performed.

### 6.1 Aircraft dynamics model

In an aircraft trajectory optimization project, it is important to get a good balance between accuracy and computational speed. One of the most influential parts in this trade-off is the modelling of the aircraft dynamics. Like in [3], a reduced-order point mass model is applied to achieve this. Since the model should work with long-haul flights, the dynamics are placed in the “World Geodetic System”, a polar coordinate system which has its origin at the center of the earth, the zero latitude  $\phi$ -axis at the equator and the zero longitude  $\lambda$ -axis near Greenwich, UK. The altitude  $z$  is denoted in meters above the surface of the earth, of which the radius  $R_e$  is estimated at  $6378 \cdot 10^3$  m. An unusual but non-influential anomaly is that heading  $\chi$  is zero in a completely eastward direction and increases counterclockwise. The aircraft states and controls are:

$$\mathbf{x} = \begin{bmatrix} \phi \\ \lambda \\ z \\ V \\ \chi \\ W \\ \gamma \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \eta \\ \dot{\gamma} \\ \mu \end{bmatrix} \quad (6.1)$$

Where  $\phi$  is the latitude [rad],  $\lambda$  is the longitude [rad],  $z$  is the altitude [m],  $V$  is the velocity [m/s],  $\chi$  is the heading angle [rad],  $W$  is the aircraft weight [N] and  $\gamma$  is the flight path angle [rad]. For the controls,  $\eta$  is the engine thrust setting between 0 and 1 [-],  $\dot{\gamma}$  is the change in flight path angle [rad/s] and  $\mu$  is the aircraft bank angle [rad]. The dynamics of the state,  $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ , are described per state variable as follows:

$$\dot{\phi} = \frac{V \cos(\gamma) \sin(\chi)}{R_e + z} \quad (6.2)$$

$$\dot{\lambda} = \frac{V \cos(\gamma) \cos(\chi)}{(R_e + z) \cos(\phi)} \quad (6.3)$$

$$\dot{z} = V \sin(\gamma) \quad (6.4)$$

$$\dot{V} = g_0 \left( \frac{T - D}{W} - \sin(\gamma) \right) \quad (6.5)$$

$$\dot{\chi} = g_0 \frac{\tan(\mu)}{V} \quad (6.6)$$

$$\dot{W} = -\dot{m}_{fuel} * g_0 \quad (6.7)$$

$$\dot{\gamma} = \dot{\gamma} \quad (6.8)$$

Here,  $g_0$  is the gravitational force of the earth (9.81 m/s<sup>2</sup>).  $T$  is defined as the thrust force,  $D$  is the drag force, and  $\dot{m}_{fuel}$  is the fuel flow of the aircraft. These parameters are obtained through the following equations:

$$T = T_{min} + \eta(T_{max} - T_{min}) \quad (6.9)$$

$$D = C_D \frac{1}{2} \rho V^2 S \quad (6.10)$$

$$\dot{m}_{fuel} = f(z, V, T) \quad (6.11)$$

Where  $T_{min}$  is the idle thrust and  $T_{max}$  is the maximum thrust [N].  $\rho$  is the air density [kg/m<sup>3</sup>] and  $S$  is the wing surface area [m<sup>2</sup>]. The total drag coefficient  $C_D$  is described as follows:

$$C_D = C_{D_0}(M) + K(M)C_L^2 \quad (6.12)$$

Here,  $C_{D_0}(M)$  and  $K(M)$  are the zero-lift drag and the induced drag coefficient as a function of Mach number  $M$ ,  $C_L$  is the lift coefficient.  $T_{max}$ ,  $T_{min}$ ,  $\rho$ ,  $C_{D_0}(M)$ ,  $K(M)$  and  $C_L$  are (also) dependent on the atmospheric conditions. To obtain these values, the atmosphere is modelled after the International Standard Atmosphere (ISA), which is a function of altitude  $z$  only.

## 6.2 Formation flight set-up

For the proposed trajectory optimization to work, the formation flight dynamics have to be implemented adequately into the aircraft dynamics model. This section describes which adaptations are made to incorporate formation flight dynamic behavior into the optimization model.

### 6.2.1 Formation flight aircraft dynamics model

In phases where a formation occurs, two or more aircraft need to be modelled. To reduce the complexity, all aircraft are modelled together as a one point system with multiple masses. This means that for the model, all states except  $W$  can be represented by the same value for all aircraft. On a practical level this is a rather big simplification, since in real life aircraft in extended formation would be separated streamwise by a significant distance. However, for the optimization it is assumed that this has little impact on the accuracy of the model, as long as the induced drag reduction factor  $C_{formation}$  is chosen realistically.

For a phase with formation size  $N$ , the state and control vectors then look as follows:

$$\mathbf{x} = \begin{bmatrix} \phi \\ \lambda \\ z \\ V \\ \chi \\ W_1 \\ W_2 \\ \vdots \\ W_N \\ \gamma \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \eta_1 \\ \dot{\gamma} \\ \mu \end{bmatrix} \quad (6.13)$$

It can be observed that in the control, only the thrust setting for aircraft 1,  $\eta_1$ , is controlled. For the other aircraft, the thrust is actually solved for by using the available settings of the lead aircraft and the ratio between the lead aircraft weight and the trailing aircraft weight:

$$T_{trail} = \left[ \frac{T - D}{W} \right]_{lead} W_{trail} + D_{trail} \quad (6.14)$$

This value is then in turn used to compute the fuel flow of the trailing aircraft. Of course, the thrust values of any aircraft in the formation cannot exceed the maximum thrust value of that respective aircraft. Therefore, the following constraint is added for every trailing aircraft:

$$\frac{T_{trail} - T_{min}}{T_{max} - T_{min}} = \eta_{trail} \leq 1 \quad (6.15)$$

## 6.2.2 Induced drag reduction factor

In Chapter 3 it was described how formation flight can be beneficial in terms of fuel consumption for trailing aircraft. Fuel is saved because there is less induced drag experienced by the trailing aircraft, so in the aircraft model, the reduction in induced drag must be modelled as well. For this, the already discussed equation for the drag coefficient  $C_D$  (Eq. 6.12) is adapted as follows:

$$C_D = C_{D_0}(M) + K(M)C_L^2 * c_f \quad (6.16)$$

Where  $c_f$  is the induced drag reduction factor for trailing aircraft in a formation. If an aircraft is either leading the formation or not in formation at all,  $c_f = 1$ . Based on the study presented in Chapter 3, values for  $c_f$  are chosen as follows:

Table 4: Values for  $c_f$  in all experiments unless stated otherwise

Aircraft	$c_f$
Leading	1
First trailing	0.75
Second trailing	0.5
Third trailing	0.5

Using Eq. 6.16 and the values above, for every aircraft  $i$  in a formation of size  $N$  the respective  $\dot{m}_{fuel_i}$  and  $W_i$  can be calculated individually by plugging them into Eq. 6.11 and 6.7 respectively. An extra point of consideration is that the assignment of the  $c_f$  values of Table 4 is fixed: Aircraft  $N$  in a phase (last weight element  $W_N$  in the state vector) will be the leading aircraft with  $c_f = 1$ . Then, aircraft  $1, 2, \dots, N-1$  will have the respective  $c_f$  values assigned from the 'First trailing' value of Table 4 onwards. So, the order of the aircraft in the formation is not a (discrete) optimizable variable.

## 6.3 Model constraints

Constraints are added to the model to limit the values which the state and control variables are allowed to take. This way unrealistic solutions are avoided, and the solver is helped with finding the correct solution. This section describes the constraints which are placed onto the model.

### 6.3.1 Time and duration constraints

The time vector used for the optimization is arbitrarily chosen in the sense that for the solution, it does not matter what the time values are, as long as the unit is consistent with the



model (e.g. seconds) and that the size of the vector is long enough to accommodate the solution. Along with the time vector, also the minimum and maximum duration is defined. Finally, for each sequence of phases which is independent (i.e. they do not have any phase linkages in common), one phase in Stage 1 (departure solo) has its initial time fixed. This prevents trajectories from spreading out over the time vector.

Time constraint:

$$-50000 \leq t \leq 60000 \text{ s}$$

Duration constraint: climb and descent solo phases;

$$600 \leq t_{f_{phase}} - t_{0_{phase}} \leq 5000 \text{ s}$$

Duration constraint: other phases:

$$5 \leq t_{f_{phase}} - t_{0_{phase}} \leq 60000 \text{ s}$$

Initial time constraint for one stage 1 phase per unlinked sequence:

$$t_{0_{phase}} = 0 \text{ s}$$

### 6.3.2 State and control limits

For each aircraft state and control, limits must be specified as well. These limits are either chosen manually to narrow down the feasible trajectory space (only transatlantic flights are considered), or they are defined by limitations of the aircraft.

State limits

$$-40 \leq \phi \leq 70^\circ$$

$$-115 \leq \lambda \leq 15^\circ$$

$$10000 \leq z \leq z_{max}$$

$$250 \leq V \leq 540 \text{ kts}$$

$$110 \leq \chi \leq 250^\circ$$

$$-20 \leq \gamma \leq 20^\circ$$

For each aircraft  $i$

$$W_{min} \leq W_i \leq W_{max}$$

Where  $W_{max}$  equals the Maximum Take-off Weight (MTOW) [N] of the aircraft, and  $W_{min}$  is defined as follows:

$$W_{min} = OEW + W_{PL} + 0.05 * W_{FC} \quad (6.17)$$

Here, OEW is the Operational Empty Weight [N],  $W_{PL}$  is the payload weight [N] and  $W_{FC}$  is the fuel capacity [N].

Control limits

$$0 \leq \eta \leq 1$$

$$-10 \leq \dot{\gamma} \leq 10$$

$$-25 \leq \mu < 25^\circ$$

### 6.3.3 Path constraints

The state and control limits by themselves are insufficient to guarantee a realistic trajectory, as they mostly have a mathematical meaning. The flight envelope or operational constraints must also be implemented, but these mostly depend on the current state of the aircraft, as well as the phase which is considered. To do this, path constraints are defined per flight stage. These path constraints are presented in Table 5.

Table 5 Overview of incorporated path constraints

	Parameter	(In)equality constraint	Notes
General constraints	Turning rate	$ \dot{\chi}  \leq 1^\circ * s^{-1}$	Limited for passenger comfort
	Speed limits	$V_{min} \leq V \leq V_{sound} * M_{MO} \left[\frac{m}{s}\right]$	Limits determined by aircraft model and atmosphere
Climbing stage (1)	Vertical speed	$0 \leq \dot{z} \leq 200 \text{ m/s}$	Prevents aircraft descent
	Energy rate	$\dot{E} = \frac{(T - D)V}{W} \geq 0$	Reduces unnecessary exchange of velocity and altitude
Descent stage (5)	Vertical speed	$\dot{z} \leq 1 \text{ m/s}$	Prevents steep aircraft climbing
	Acceleration	$\dot{V} \leq 0 \text{ m/s}^2$	Prevents aircraft acceleration during descent
(Formation) cruise stage (2/3/4)	Vertical speed	$ \dot{z}  \leq 1 \text{ m/s}$	Prevents steep aircraft climbing or descent
	Acceleration	$ \dot{V}  \leq 1 \text{ m/s}^2$	For passenger comfort
	Energy rate	$\dot{E} = \frac{(T - D)V}{W} \geq 0$	Reduces unnecessary exchange of velocity and altitude
	Engine thrust setting of each trailing aircraft $i$	$0 \leq \eta_i \leq 1$	Ensures that no aircraft in the formation (if a formation is flown) exceeds its thrust limit

### 6.3.4 Phase linkages

The trajectory optimization problem is a multi-phase problem, which consists of multiple continuous trajectory phases which are linked together at discrete switching points. The linkage between two phases is to be defined such that the final state values of the left phase  $\mathbf{x}_{f,l}$  equal the initial state values of the right phase  $\mathbf{x}_{0,r}$ . However, in this case the phase switching point also represents a change in formation size, and therefore also a change in the weight vector size. The linkage constraints are then defined as follows:

If formation size increases from the left phase to the right phase (formation buildup):

$$\begin{bmatrix} \phi_{0,r} \\ \lambda_{0,r} \\ z_{0,r} \\ V_{0,r} \\ \chi_{0,r} \\ W_{0,r}^L \\ \gamma_{0,r} \end{bmatrix} - \begin{bmatrix} \phi_{f,l} \\ \lambda_{f,l} \\ z_{f,l} \\ V_{f,l} \\ \chi_{f,l} \\ W_{f,l}^L \\ \gamma_{f,l} \end{bmatrix} = \mathbf{0} \quad (6.18)$$

If formation size decreases from the left to the right phase (formation breakdown):

$$\begin{bmatrix} \phi_{0,r} \\ \lambda_{0,r} \\ z_{0,r} \\ V_{0,r} \\ \chi_{0,r} \\ W_{0,r}^R \\ \gamma_{0,r} \end{bmatrix} - \begin{bmatrix} \phi_{f,l} \\ \lambda_{f,l} \\ z_{f,l} \\ V_{f,l} \\ \chi_{f,l} \\ W_{f,l}^R \\ \gamma_{f,l} \end{bmatrix} = \mathbf{0} \quad (6.19)$$

Where  $W^L$  and  $W^R$  are the aircraft weights of the aircraft present in the left and right phase respectively.

#### 6.4 Aircraft model (Boeing 747-400)

For this optimization case study, a model of a Boeing 747-400 aircraft is used unless stated otherwise. The specifications of this aircraft are listed in Table 6.

Table 6: Boeing 747-400 parameters

Parameter	Description	Value
S	Wing surface area [m <sup>2</sup> ]	541.16
MTOW	Maximum Take-Off Weight [kg]	362,874
OEW	Operational Empty Weight [kg]	178,756
MLW	Maximum Landing Weight [kg]	260,362
MZFW	Maximum Zero Fuel Weight [kg]	242,672
Fuel capacity	Maximum weight of fuel [kg]	163,396
Payload capacity	Maximum weight of payload [kg]	63,917
$z_{max}$	Maximum altitude [m]	13,747
$M_{MO}$	Maximum Mach number [-]	0.9
$V_{min}$	Minimum velocity [kts]	195

For this aircraft, the parameters  $C_{D_0}(M)$  and  $K(M)$  are represented as follows [3]:

$$C_{D_0}(M) = -0.001M^2 + 0.0006M + 0.0125 \quad (6.20)$$

$$K(M) = 0.0975M^2 - 0.0846M + 0.0705 \quad (6.21)$$



Figure 16: Boeing 747-400

## 6.5 Bréguet range equation for formation flight

For this trajectory optimization, it is chosen to represent the aircraft dynamics with a reduced order point-mass model. To reduce the computational intensiveness, this model can be replaced with an analytic equation for the cruise formation phase. In normal aircraft trajectory calculations, the Bréguet range equation can be applied to obtain the fuel usage as a function of the required range, velocity and altitude. However, the traditional Bréguet range equation is not applicable to formation flight. Therefore M. Voskuijl presented a modified form of the Bréguet range equation which is adapted to work with formation flight under realistic operating conditions (constant Mach number and constant altitude at transonic speeds) [31]. This modified equation incorporates all factors which determine the difference in fuel usage as a result of the induced drag benefit associated with flying in formation.

### 6.5.1 Equations

The modified Bréguet range equation, rewritten to express the final weight  $W_f$ , is presented below:

For each aircraft in the formation  $N$  with induced drag reduction factor  $C_f^N$ :

$$W_f^N = \frac{1}{2} \gamma p M^2 S \left[ C_L^* - \sqrt{\frac{C_D^*}{C_f^N K(M)}} \tan \left\{ \frac{C_0 R (1 + C_M M) * \sqrt{C_f^N K(M) C_D^*}}{a M} + \tan^{-1} \left( \sqrt{\frac{C_f^N K(M)}{C_D^*}} \frac{\left( \frac{1}{2} \gamma p M^2 S C_L^* - W_0^N \right)}{\frac{1}{2} \gamma p M^2 S} \right) \right\} \right] \quad (6.22)$$

Where  $C_0$  and  $C_M$  are empirical constants for the aircraft engine performance.  $\gamma$  here indicates the air specific heat ratio (1.4),  $p$  is the air pressure [kg/m<sup>3</sup>] and  $a$  is the speed of sound [m/s].  $C_f$  is an array of induced drag reduction factors  $c_f$ .  $R$  [m] is the distance flown, but this parameter is not part of the state of the aircraft model. To solve this,  $R$  is found as a function of  $\phi$ ,  $\lambda$  and  $z$  using the 'haversine' formula:

$$R = 2 * (z + R_e) * \text{atan2}(\sqrt{k}, \sqrt{1-k}) \quad (6.23)$$

$$k = \sin^2\left(\frac{\phi_f - \phi_0}{2}\right) + \cos\phi_0 \cos\phi_f \sin^2\left(\frac{\lambda_f - \lambda_0}{2}\right) \quad (6.24)$$

With this method, the final weight at the end of a phase is a function of the change in latitude and longitude. Since all other states in the considered aircraft dynamics model are supposed to stay equal, an entire cruise phase can be replaced with an evaluation of the modified Bréguet range equation, reducing the number of nodes and intervals significantly.

### 6.5.2 Implementation

The modified Bréguet equations are to be implemented in such a way that the cruise phase with the largest formation size is evaluated. To achieve this, in GPOPS this phase is reduced to a ‘dummy phase’ with 2 nodes and a fixed duration of 5 seconds. This dummy phase is linked as usual to its preceding phases. Then, the modified Bréguet range equation is implemented in the phase linkages which link this dummy phase to its succeeding phases as follows:

$$\begin{bmatrix} \phi_{0,r} \\ \lambda_{0,r} \\ z_{0,r} \\ V_{0,r} \\ \chi_{0,r} \\ W_{0,r}^R \\ \gamma_{0,r} \end{bmatrix} - \begin{bmatrix} \phi_{f,l} \\ \lambda_{f,l} \\ z_{f,l} \\ V_{f,l} \\ \chi_{f,l} \\ W_{f,l}^R \\ \gamma_{f,l} \end{bmatrix} = \begin{bmatrix} \text{free} \\ \text{free} \\ 0 \\ 0 \\ \text{free} \\ f_{breguet}(W_{f,l}^R, \phi_{f,l}, \lambda_{f,l}, z_{f,l}, V_{f,l}, \phi_{0,r}, \lambda_{0,r}, C_f) - W_{f,l}^R \\ 0 \end{bmatrix} \quad (6.25)$$

Where  $f_{breguet}$  represents the modified Bréguet range equation presented in Equation Eq. 6.22 and  $C_f$  is the vector of all respective induced drag reduction factors  $c_f$ . GPOPS is then tasked with choosing  $\phi_{f,l}, \lambda_{f,l}, \phi_{0,r}$  and  $\lambda_{0,r}$  such that the original objective value is minimized.

In practice, the reduced number of nodes and the associated computational efficiency benefit is found to be negated by the fact that GPOPS appears to have a great difficulty with linkage constraints where state variables are left free. Although during testing realistic solutions are still obtained, the convergence behavior is poor. Therefore, this method is only used in Experiment 5 (Section 7.5) for demonstration purposes. Another drawback is that time is not part of the model during this phase, leading to incorrect flight time values.

## 6.6 Optimization parameters

For the developed hybrid trajectory optimization tool, several parameters influence the behavior of the optimization algorithm and therefore also the outcome. In general, values for these parameters are a trade-off between accuracy and computation time. The relevant parameters and their chosen values are presented in this section.

### 6.6.1 Number of intervals, nodes per interval

These parameters determine the resolution of the solution trajectories. The sum of these two parameters defines the total number of collocation points. The number of nodes per interval determines the order of the fitting polynomial for satisfying the dynamics model at each collocation point. In this thesis, a resolution of 25 intervals per phase with 2 nodes per interval (linear fit) is chosen as this keeps computation times low enough while providing sufficiently accurate trajectories.

### 6.6.2 Feasibility tolerance, optimality tolerance

Based on these parameters SNOPT determines to which degree the feasibility and optimality conditions must be satisfied before it can stop the optimization. A lower value means that a more accurate result is required, resulting in longer computation times. In this set-up,  $10^{-6}$  is chosen for feasibility and  $10^{-4}$  for optimality.

### 6.6.3 Branch-and-bound cutoff margin

This parameter describes to which percentage the solution of a node in the branch-and-bound solution tree may exceed the objective value of the best node on the same tree level. If the margin is exceeded, the subtree will be cut off. In this set-up, the cutoff margin is set at 0.5%.

### 6.6.4 Iteration limit

This parameter sets the number of iterations which SNOPT is allowed to perform before an optimization run is deemed infeasible due to lack of convergence. This value is set at 70000.

## 6.7 Assumptions and simplifications for the experiments

In the case studies performed for this thesis, several assumptions and simplifications are made which could cause significant deviations of the trajectory optimization results from a real-world scenario. These are presented in this section.

### 6.7.1 Atmosphere and wind

First of all, as mentioned in Section 6.1, the atmosphere is modelled after the International Standard Atmosphere (ISA), whereas in real life atmospheric conditions could

be significantly different. In addition, it is also assumed that there is no wind present, even though there are always wind effects present in real world transatlantic flights. Both of these aspects have a significant impact on the 4D trajectory of all modelled aircraft.

### **6.7.2 Trajectory Based Operations**

The optimization is performed with a Trajectory Based Operations (TBO) implementation, meaning that the aircraft are free to move through the available airspace without any limitations posed by ATC infrastructure. Aircraft do not have to follow certain waypoints but can rather follow any route it wants. This is in real life impossible, but the movement towards TBO is clear as described in [32]. Thus, the experiments in this thesis focus on the expected development where TBO becomes reality.

### **6.7.3 Constant induced drag reduction factor**

Regarding the induced drag during formation flight, it is assumed that this is reduced for trailing aircraft by a constant value. In real life however, it is more likely that the reduction in induced drag is varying constantly, as a result of non-static relative positioning of aircraft as well as disturbing factors such as wind. The induced drag reduction factor values should thus rather represent a good estimate for the average aerodynamic benefit.

### **6.7.4 Start and landing**

For all experiments, the initial and final conditions for altitude and speed are set to 1000 ft. and 250 knots respectively. This implies that landing and take-off phases are not simulated, but rather just the climb and descent. Also, the start and end locations are taken to be the official coordinates of the origin and destination airports, independent from the chosen runways and departure/arrival routes.

### **6.7.5 Fuel reserve requirements**

Optimizing for fuel burn automatically leads to a solution where the aircraft will carry the minimum amount of fuel possible upon arrival. In the experiments, it is allowed for aircraft to take as little fuel as possible to achieve this. However, such an approach would in real life not be very robust, since a failed formation attempt could then lead to fuel capacity issues for aircraft which had 'counted' on the fuel benefits but did not get it. The results of the experiments therefore represent a theoretical optimum, which should be corrected for manually before the calculated trajectories are practically viable and safe.

### **6.7.6 Payload**

In all experiments, it is assumed that the payload weight equals the maximum payload capacity. Since all experiments are performed with a Boeing 747-400 model, the payload will always be 63,917 kg.






## 7 Case Studies: Results

This chapter presents and discusses the results of various multi-aircraft formation flying scenarios which were evaluated using the developed hybrid optimal control approach. In these experiments, each flight is assigned a different trajectory color to keep a clear distinction between them. For formation phases, again different colors are used for clarity. To make a useful analysis about the benefit of formation flying, all results are also to be compared to the results of an all solo flight scenario.

### 7.1 Experiment 1: Model validation with three-aircraft transatlantic scenario

The first experiment is designed to validate the developed branch-and-bound model by comparing its outcome to a full enumeration of the solution space. A three-aircraft case is considered with transatlantic flights from Europe to the United States. Each flight has unique departure and arrival airports, but given that their transatlantic trajectory is rather similar, formation flying possibilities still arise. These flights are presented in Table 7. All non-intermediate phase structure possibilities were evaluated manually, and consecutively the branch-and-bound algorithm was used to find the optimal phase structure.

Table 7: Flights for Experiment 1

Flight	Trajectory color	Aircraft type	Origin	Destination
1		Boeing 747-400	London (LHR)	Atlanta (ATL)
2		Boeing 747-400	Madrid (MAD)	New York City (JFK)
3		Boeing 747-400	Amsterdam (AMS)	Boston (BOS)

In Table 8, the cost values expressed as the cumulative fuel consumption of all flights are presented for each possible phase structure. In Figure 17, the evaluated branch-and-bound solution tree nodes are presented with their respective cost values.

It can be seen that the branch-and-bound algorithm finds the same objective value ( $198,57 * 10^3$  kg) and optimal structure as a total enumeration of the tree, while the number of evaluated nodes is reduced. This proves the validity of the branch-and-bound approach for this experiment. The optimal phase structure is:

$$[ \quad 0 \quad 13 \quad 123 \quad 12 \quad 0 \quad ]$$

The associated cost value of  $198,57 * 10^3$  kg is a 4,74% fuel saving compared to the solo flight trajectories. The trajectory diagrams of this phase structure are presented in Figure 18 and Figure 19.

Table 8: Cost value of all phase structure combinations for Experiment 1

Structure ID	Formation phase structure					Cost value: cumulative fuel consumption [ $10^3$ kg]
1	0	12	123	12	0	200,46
2	0	13	123	12	0	198,57
3	0	23	123	12	0	200,46
4	0	12	123	13	0	200,87
5	0	13	123	13	0	198,96
6	0	23	123	13	0	200,88
7	0	12	123	23	0	200,87
8	0	13	123	23	0	198,98
9	0	23	123	23	0	200,87
10		0	12	0		204,63
11		0	13	0		202,18
12		0	23	0		209,46
13			0			208,39

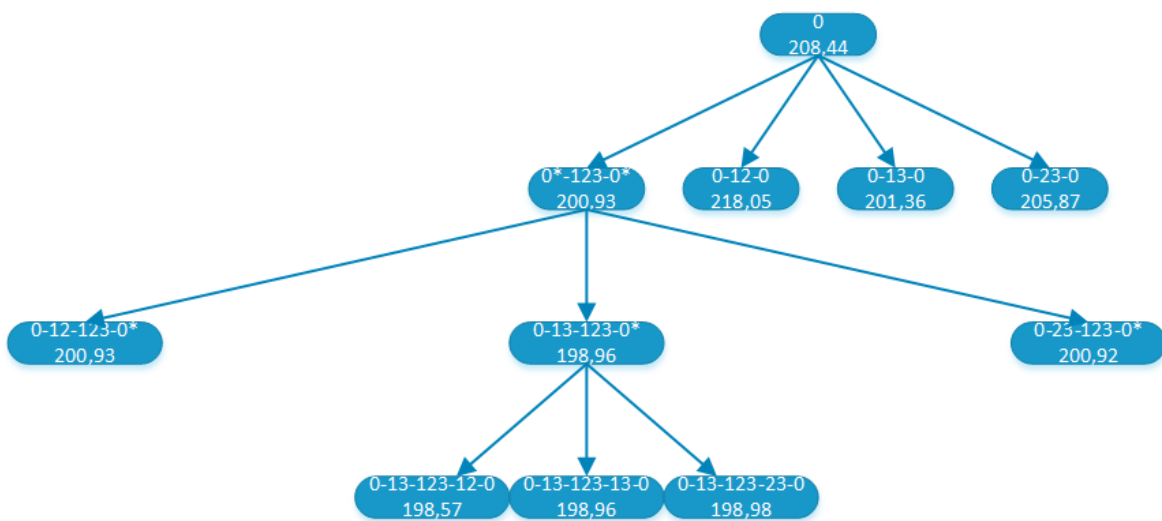


Figure 17: Experiment 1 branch-and-bound evaluated solution tree with cost values per node [ $10^3$  kg]

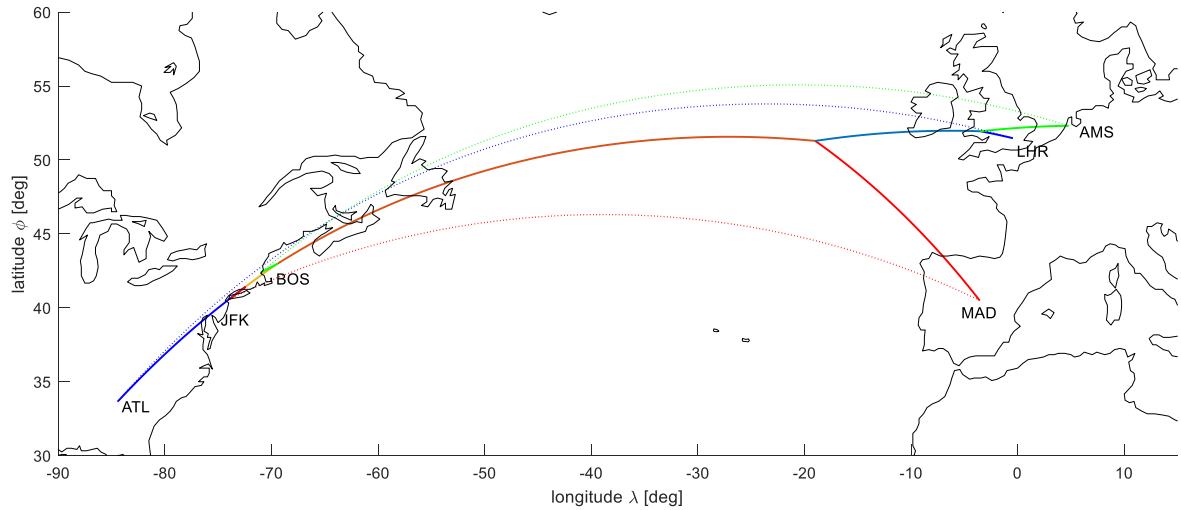


Figure 18: Horizontal trajectory of Experiment 1: solo (dotted lines) vs. formation (solid lines)

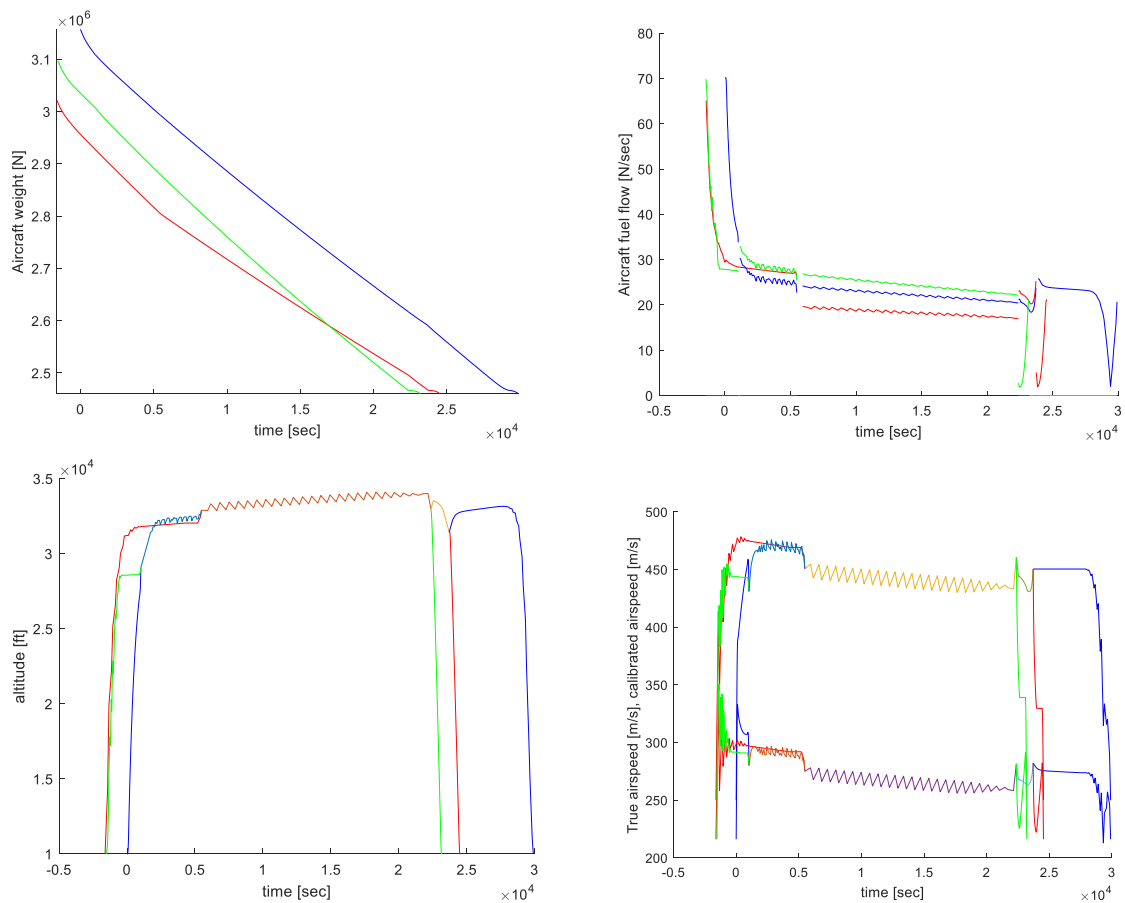


Figure 19: Experiment 1 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right).

In Figure 18, it can be observed that Flight 2 makes a relatively big detour to join the formation. This is explained by the fact that this is the aircraft experiencing a 50% induced drag reduction, making the detour worthwhile. In Figure 19b, c and d, a general observation is that jagged lines are present. These are caused by the fact that the model is free to exchange velocity and altitude without fuel burn penalty, in combination with a low resolution of 25





intervals with 2 nodes per interval (linear fit). Also, it is clear that the velocity during cruise phases is lower than during solo phases.

In this experiment, the savings in number of evaluated tree nodes are minor, since only two nodes less were evaluated effectively. However, the complete enumeration required manually generated initial guesses. The branch-and-bound algorithm is still significantly more convenient to use, since it evaluates the solution space automatically with better convergence.

## 7.2 Experiment 2: Four-aircraft transatlantic formation

In this experiment, an extra flight is added to the flights of Experiment 1 such that four fictional transatlantic flights are modelled which are candidates for a formation. This experiment is expected to take the most advantage of the branch-and-bound algorithm, since there are hundreds of combinatorial solutions possible which the algorithm is expected to narrow down to the most promising ones and eventually select the best solution. The modelled flights are as presented in Table 9:

Table 9 Flights for Experiment 2

Flight	Trajectory color	Aircraft type	Origin	Destination
1		Boeing 747-400	London (LHR)	Atlanta (ATL)
2		Boeing 747-400	Madrid (MAD)	New York City (JFK)
3		Boeing 747-400	Amsterdam (AMS)	Boston (BOS)
4		Boeing 747-400	Rome (FCO)	Toronto (YYZ)

In Table 10 the trajectory optimization results for the solo and the formation case are presented together. The horizontal trajectories for both cases are also presented in Figure 20. In Figure 21, several other trajectory parameters are presented for the formation case only. Full trajectory diagrams for the solo case can be found in Appendix C.

Table 10: Experiment 2 fuel consumption and flight time results, solo vs formation.

Flight	Fuel consumption [ $10^3$ kg]			Flight time [hh:mm:ss]		
	solo	formation	difference	solo	formation	difference
1	79.024	71.626	-7.398 (-9.36%)	07:55:23	08:21:22	00:26:00 (+5.47%)
2	66.165	57.362	-8.803 (-13.30%)	06:48:01	07:23:15	00:35:14 (+8.64%)
3	63.497	50.388	-13.109 (-20.64%)	06:33:39	06:58:51	00:25:13 (+6.41%)
4	83.338	86.672	+3.334 (+4.00%)	08:17:13	08:53:07	00:35:54 (+7.22%)
Total	292.026	266.050	-25.976 (-8.90%)	29:34:15	31:36:36	02:02:21 (+6.90%)

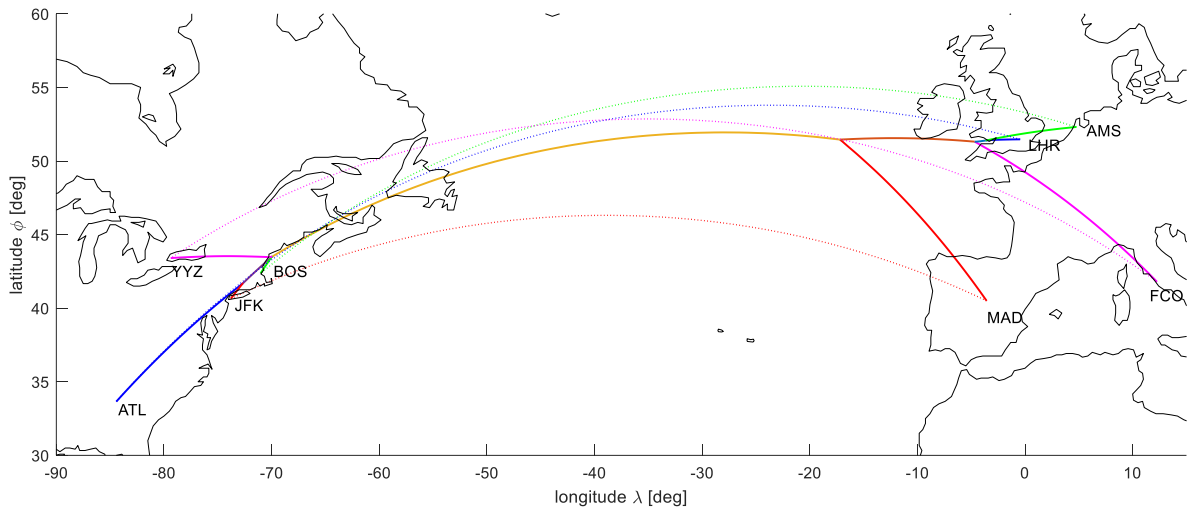


Figure 20: Horizontal trajectory of Experiment 2: solo (dotted lines) vs. formation (solid lines)

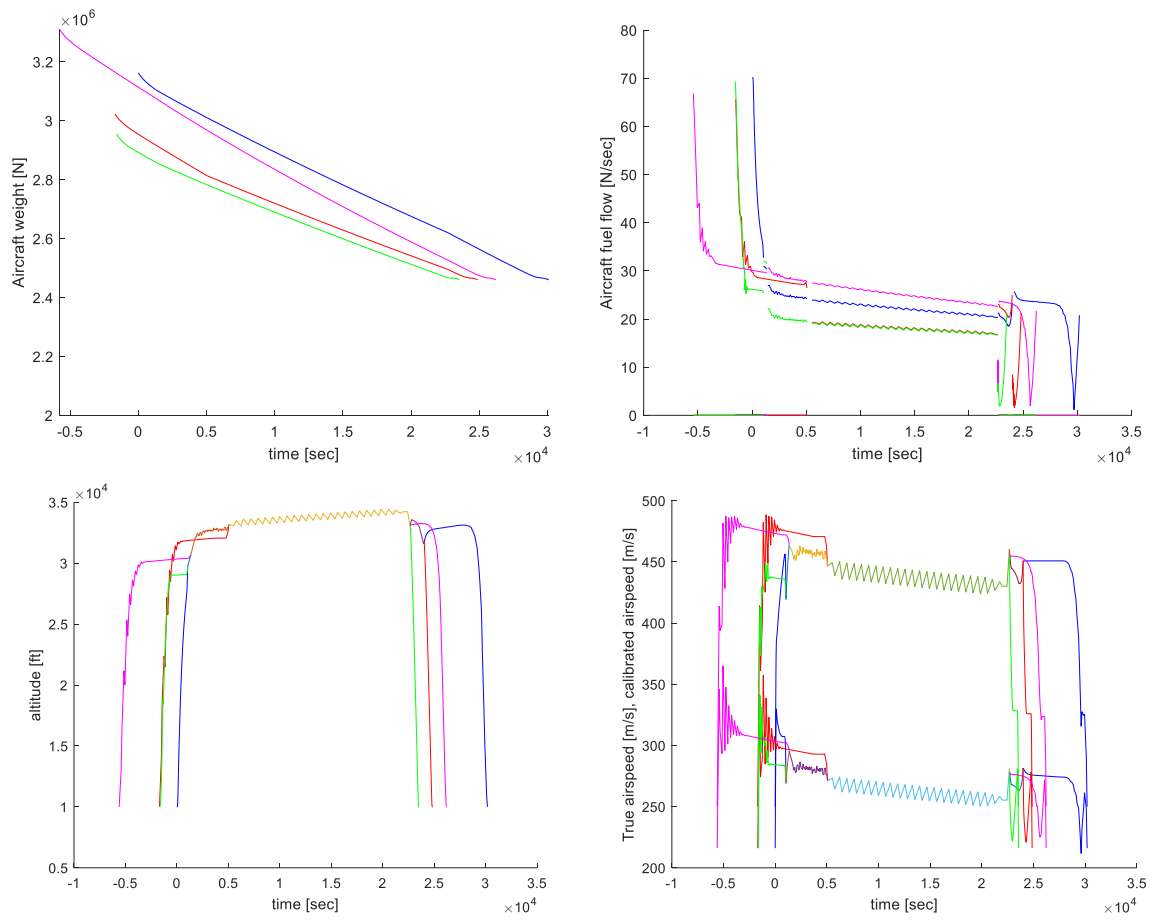


Figure 21: Experiment 2 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right).

The optimal phase structure given by the branch-and-bound algorithm is:

[ 0 34 134 1234 123 12 0 ]

However, it can be observed from Figure 20 that aircraft 1, 3 and 4 join each other roughly at the same time. Similarly, aircraft 3 and 4 also leave the formation at roughly the same time. To reach this optimal result, the branch-and-bound algorithm evaluated 35 different phase structures in the solution tree, which is 10,1% of the total amount of 347 possible phase structures.

The four evaluated flights in this experiment demonstrate that joining in formation can save a significant amount of fuel (25.976 kg / 8.90%) compared to their respective solo flights. Even for Flight 2 (MAD-JFK), which has to fly a relatively large detour to join the formation, the fuel savings are beneficial enough to do it. During the formation buildup, the staged approach where one aircraft joins after another is clearly the most optimal, since it allows for Flight 2 to join the formation much later than the other aircraft. However, the model appears to strongly prefer three-aircraft formation phases over two-aircraft formation phases, since the phase duration of the two-aircraft phases is relatively short during both the buildup and the breakdown of the formation. This can be readily explained by the significant induced drag reduction benefit assigned to the additional trailing aircraft in a three-aircraft formation.

Regarding flight time, sacrifices are made to achieve the most fuel-efficient result. However, the additional flight time never exceeds 36 minutes, which could be deemed acceptable for a transatlantic flight. Of course, the additional flight times can be attributed to the fact that all aircraft have to compromise on their shortest distance route to meet the other aircraft in a formation. However, the cruise speed during the formation is also significantly lower than during the solo phases (Figure 21 d). It appears that a lower induced drag also results in a lower optimal cruise speed with respect to fuel use.

Due to the fact that the lay-out of the formation in terms of relative position is not optimized for, the overall solution could have been more optimal if a different aircraft was chosen as a leading aircraft. In this experiment, aircraft 4 leads the formation and thus receives no fuel benefit. However, it is flying the longest flight, and therefore it is one of the heavier aircraft during the formation. In previous research it was found that in formation flight with identical aircraft models, the lightest aircraft should lead the formation [3], which in this case should have been aircraft 3.

The optimization problem is solved in 18437 seconds, or 5:07:07 using a quad-core CPU. The average calculation time per node is 1714 seconds. However, since the optimization is running multi-threaded, the effective calculation time per node is lower. These values indicate that the hybrid optimal control approach is still within practical limits for a four-aircraft scenario, but still a significant amount of time is needed to obtain the solution. A real-life implementation might require a faster method, especially when 'on-the-fly' iterations are required due to changing circumstances.

### 7.3 Experiment 3: Sensitivity analysis with less induced drag benefit

Even though the standard values chosen for the induced drag reduction factor, presented in Section 6.2.2, are based on real experimental results, there is a lot of variance possible in the induced drag benefit due to disturbances (e.g. gust winds) or lateral positioning errors. Therefore, to analyze the sensitivity of the trajectory results of Experiment 2, the same experiment is repeated with a more pessimistic induced drag reduction factor. The values chosen for this experiment are presented in Table 11. Otherwise, the experimental input is identical to Experiment 2. The results of Experiment 3 are presented in Table 12, Figure 22 and Figure 23.

Table 11: Values for  $c_f$  in Experiment 3

Aircraft	$c_f$
Leading	1
First trailing	0.875
Second trailing	0.75
Third trailing	0.75

Table 12: Experiment 3 fuel consumption and flight time results, solo vs formation.

Flight	Fuel consumption [ $10^3$ kg]			Flight time [hh:mm:ss]		
	solo	formation	difference	solo	formation	difference
1	79.024	75.283	-3.741 (-4.73%)	07:55:23	08:09:46	00:14:24 (+3.03%)
2	66.165	63.426	-2.739 (-4.14%)	06:48:01	07:04:51	00:16:50 (+4.13%)
3	63.497	58.021	-5.476 (-8.62%)	06:33:39	06:44:57	00:11:19 (+2.87%)
4	83.338	85.137	+1.799 (+2.16%)	08:17:13	08:32:58	00:15:45 (+3.17%)
Total	292.026	281.867	-10.159 (-3.48%)	29:34:15	30:32:33	00:58:18 (+3.29%)

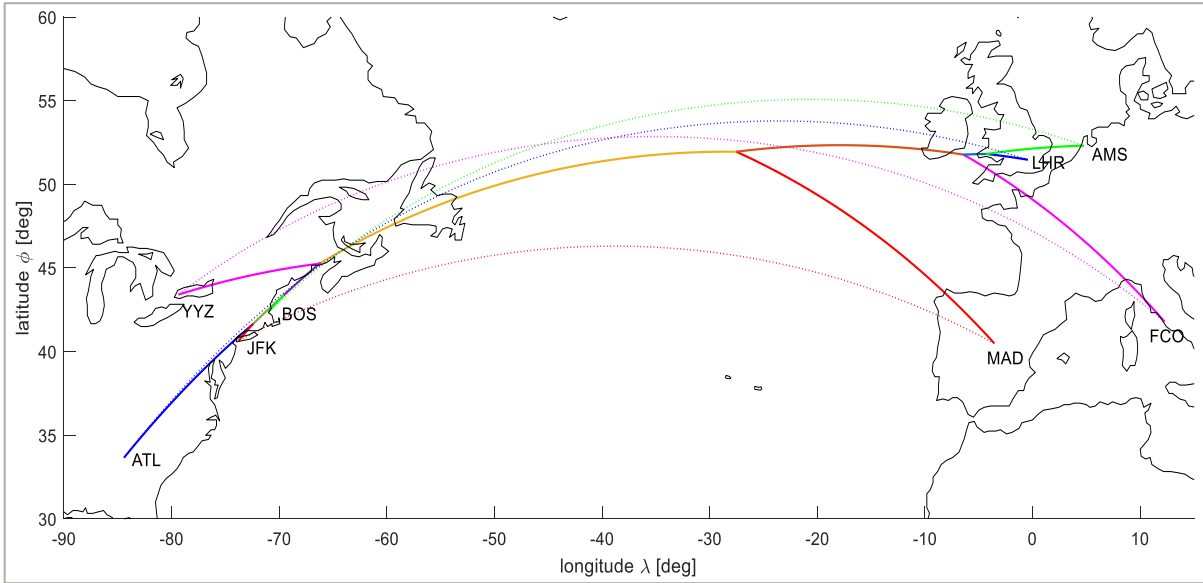


Figure 22: Horizontal trajectory of Experiment 3: solo (dotted lines) vs. formation (solid lines)

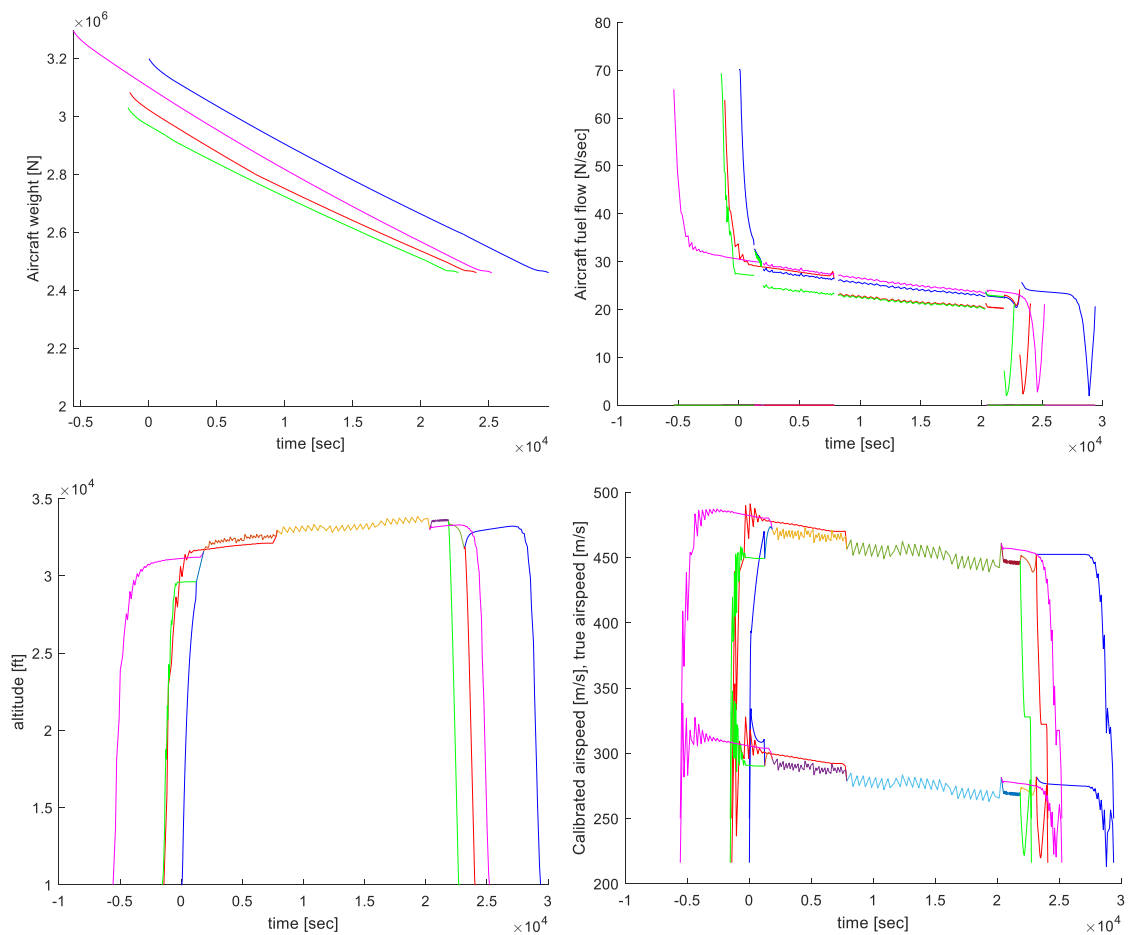


Figure 23: Experiment 3 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right).



The optimal phase structure given by the branch-and-bound algorithm is:

[ 0 13 134 1234 123 12 0 ]

Which is slightly different from Experiment 2. To reach this optimal result, the branch-and-bound algorithm evaluated 56 different phase structures in the solution tree, which is 16,1% of the total amount of 347 possible phase structures.

With a more pessimistic induced drag reduction scheme, the optimal trajectory still shows roughly the same characteristics when compared to Experiment 2. However, some major differences can be noted: Aircraft 2, which could have been expected to not join the formation at all due to the relatively marginal benefit, still joins the formation but at a much later stage. Moreover, aircraft 1 and 3 join each other early while the meeting with aircraft 4 is delayed a bit. The formation buildup and breakdown are more gradual, with only one aircraft joining or leaving at a time. This results in a slightly different optimal phase structure.

Regarding fuel savings, the amount of fuel saved is reduced by more than half to 10.159 kg (3.48%). However, this is still a significant fuel benefit, and the fact that the phase structure has not changed dramatically to achieve this shows that the chosen formation flight approach is robust enough. On the other hand, flight times are relatively beneficial, with the total increased flight time staying under an hour at 00:58:18 (+3.29%). The cruise velocity is higher in this experiment, and the model tends to follow a more direct route since the benefit of joining the formation is less strong.




The fuel saving figure is very sensitive to changes in induced drag reduction, but this does not result in significant trajectory structure changes. It might be of interest to test a wide range of induced drag reduction schemes for the model's sensitivity, but in a four-aircraft case this would demand too much computation time to be practically achievable. A induced drag reduction factor sensitivity analysis with a range of different values for a two aircraft scenario has been performed by [3].

#### 7.4 Experiment 4: Joint-Venture scenario: KLM, Air France, Virgin Atlantic

A practical implementation of formation flight from an airline perspective is generally challenging, because a good amount of cooperation between airlines who would plan the formation flight is necessary. Since they are likely to be competitors, making a formation arrangement which everyone agrees to would not be easy to achieve. However, there are also many airlines who cooperate through alliances, joint ventures or a joint shareholder interest. KLM and Air France are already operating under the same company (Air France-KLM Group), and share a lot of their resources to achieve higher efficiency. In August 2017, Air France-KLM Group also took a share of 31% in Virgin Atlantic, a long-haul airline which operates mostly from London Heathrow Airport (LHR) [33]. With an increased cooperation expected between AF-KLM Group and Virgin Atlantic, it is of interest to see if a flight of each airline in their current schedule could be operated in formation flight without changing the departure times

significantly. Three flights to New York JFK, each operated daily, were found which depart roughly at the same time [34]:

Table 13: Flights for Experiment 4

Flight no.	Trajectory color	Departure time	Origin	Destination
KL 641		13:20	Amsterdam (AMS)	New York City (JFK)
AF 6		13:40	Paris (CDG)	New York City (JFK)
VS 45		14:00	London (LHR)	New York City (JFK)

This set of flights is taken as a case study to evaluate how much the formation flight benefit would be if the departure times are fixed to the times stated in Table 13. Since all flights have the same destination, it is expected that the formation can be maintained until descent. In real life, the aircraft which are used are a Boeing 777, an Airbus A380 and a Boeing 787. However, due to a lack of available flight models for these aircraft, all flights are modelled with a Boeing 747-400. Similar to Experiment 2, the solo versus formation trajectory optimization results are presented in Table 14, with the trajectories presented in Figure 24 and Figure 25. Solo trajectory diagrams are further presented in Appendix D.

Table 14: Experiment 4 fuel consumption and flight time results, solo vs formation.

Flight	Fuel consumption [ $10^3$ kg]			Flight time [hh:mm:ss]		
	solo	formation	difference	solo	formation	difference
KL 641	67.302	61.452	-5.850 (-9.52%)	07:31:15	06:54:01	00:37:14 (+8.25%)
AF 6	67.198	51.949	-15.248 (-29.35%)	07:11:28	06:53:22	00:18:06 (+4.19%)
VS 45	63.453	64.189	+0.736 (+1.15%)	06:51:15	06:32:54	00:18:21 (+4.46%)
Total	197.954	177.591	-20.362 (-11.47%)	21:33:58	20:20:17	01:13:42 (+5.70%)

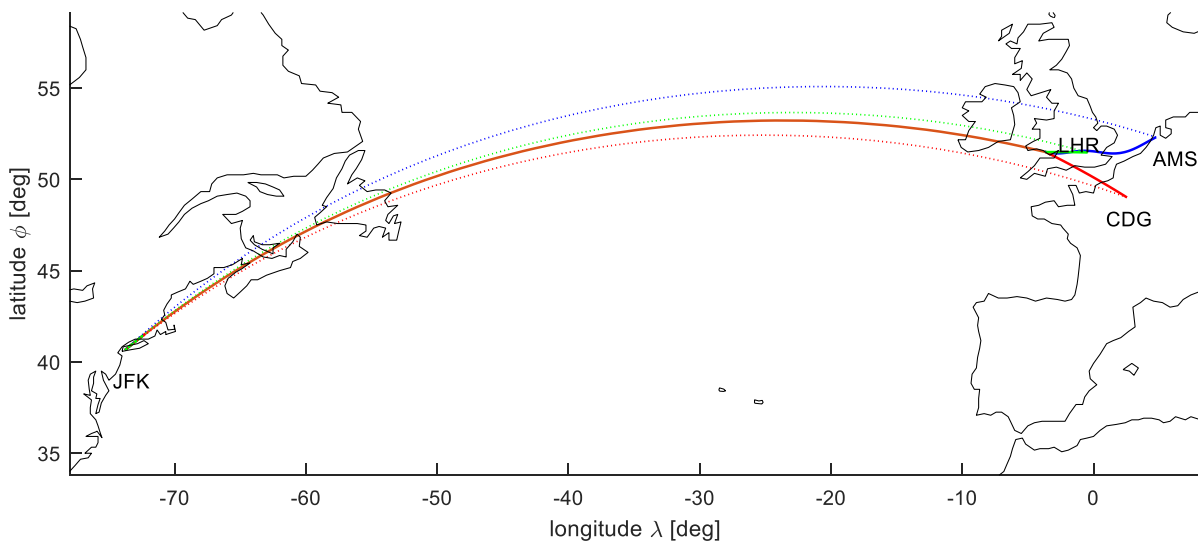


Figure 24: Horizontal trajectory of Experiment 4: solo (dotted lines) vs. formation (solid lines)

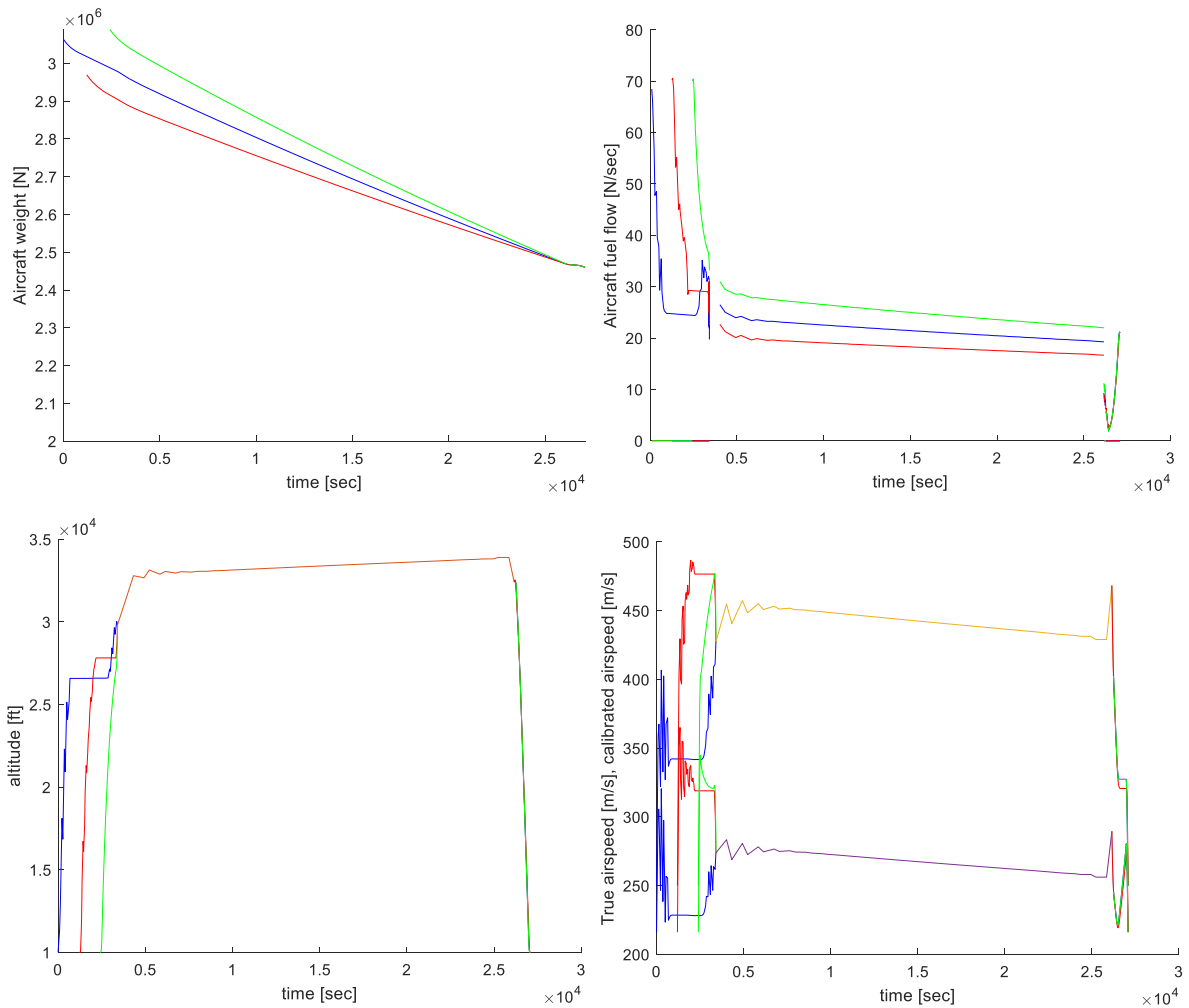


Figure 25: Experiment 4 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right).

The optimal phase structure given by the branch-and-bound algorithm is:

[ 0 12 123 13 0 ]

Also in this case, all aircraft split up at around the same time, so the '13' formation phase is of minimal duration. This is to be expected given that all aircraft have the same destination. However, this time also the rendezvous point is essentially such that all aircraft join at the same time. In the branch-and-bound algorithm, the buildup as well as the breakdown phase yield essentially the same outcome for all phase structures. This meant that all 17 possible phase structures were evaluated, because the cut off conditions for tree branches were never met. Since this is only a three-aircraft problem, the calculation time was still limited at 3844 seconds.

This experiment clearly demonstrates the feasibility of formation flying from a scheduling perspective. All three flights share a significant common path in their existing planning, and require no departure time adjustment to achieve the fuel benefits associated with formation flying. However, it can be observed that flight KL 641 flies at lower speeds and with a detour prior to joining the formation, since it needs to ‘wait’ for the other two flights to catch up. Delaying the departure time of KL 641 would thus have further increased the efficiency of this set of flights. Flight AF 6 and VS 45 are however well aligned in their current schedule.

A combined fuel savings of 25.976 kg / 8.90% is achieved, which is rather high for a three-aircraft formation. Percentage wise this savings exactly matches the savings of Experiment 2, despite the higher average induced drag during formation. This can be explained with the fact that the solo trajectories are already much more aligned than the flights of Experiment 2, making for relatively smaller detours and a reduced associated fuel burn. The total flight time increase is limited at 01:13:42 / 5.70%, and seems to be mostly caused by the lower cruise speed attained during the formation flight. The detour made by KL 641 also contributes significantly, with the flight time increase of this flight being roughly twice as high as the other flights.

Similar to experiments 1 and 2, a point can be made about the question which aircraft is in which position of the formation. Here it was deliberately chosen to make flight VS 45 the leading aircraft, since it has the shortest flight distance and therefore the lowest fuel requirement solo. However, due to the fact that it is leading the formation, it still had the highest initial weight of the three flights which were performed, making it the heaviest aircraft while leading the formation in the end. A vicious circle thus arises where the leading aircraft is by definition the heaviest even though it is suboptimal from a formation flying perspective. This situation is created because each aircraft is allowed to take as little fuel on board as needed to perform the flight with the chosen trajectory, meaning that flights KL 641 and AF 6 take less fuel on board than when they would have flown solo. Adding a constraint which forces all flights to take the fuel which they would have needed to perform the solo flights would solve this problem for the most part. However, the situation might still arise that during a formation phase, a trailing aircraft becomes lighter than the leading aircraft. Then, switching the positions of these aircraft would be more optimal. This possibility is however not considered for this simulation.

Regarding the numerical performance, the computation time stayed within practical limits at 3844 seconds. However, the evaluated solution tree reveals that the optimal result could have been obtained quicker if the branch-and-bound algorithm had a means to recognize the fact that the formation breakdown is virtually identical for all phase structures. 9 nodes could have then been removed, which is more than half of the total amount of nodes of this relatively simple three-aircraft case. In effect, the branch-and-bound algorithm did not sort any time saving effect since all nodes in the solution tree were evaluated.

## 7.5 Experiment 5: Breguet vs. non-Breguet for cruise

In this experiment, Experiment 4 is repeated but this case the modified Bréguet range equations (Section 6.5) are activated for the 3-aircraft formation cruise phase. The results of this experiment are presented in Table 15. The associated trajectory diagrams are presented in Figure 26 and Figure 27.

Table 15: Experiment 5 fuel consumption results

Flight	Fuel consumption [ $10^3$ kg]		
	Experiment 4	Experiment 5	difference
KL 641	61.452	61.312	-0.140 (-0.25%)
AF 6	51.949	52.196	+0.247 (+0.48%)
VS 45	64.189	64.292	+0.103 (+0.16%)
Total	177.59	177.80	+0.210 (+0.12%)

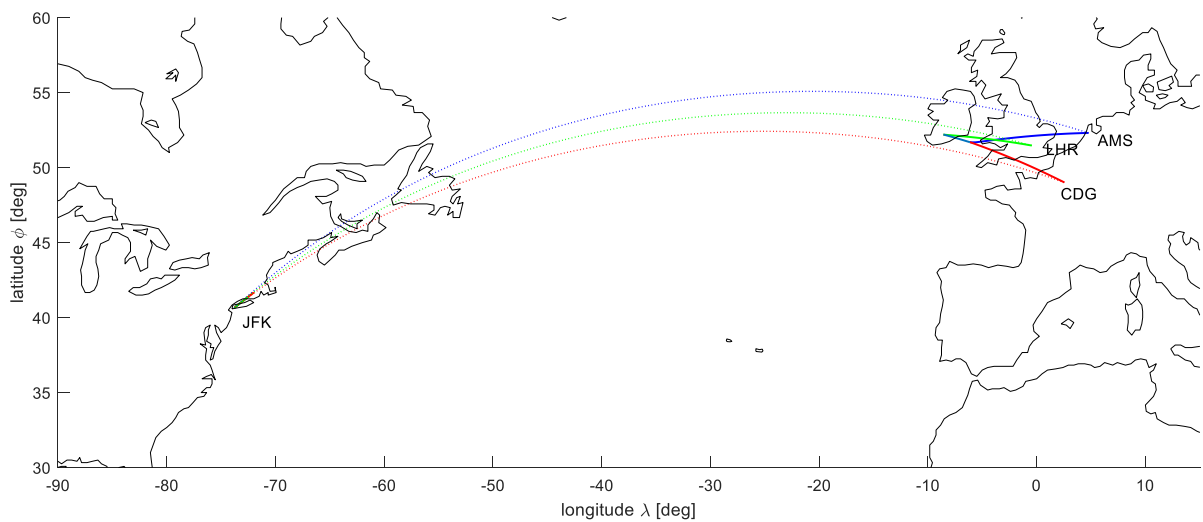


Figure 26: Horizontal trajectory of Experiment 4: solo (dotted lines) vs. formation with Bréguet (solid lines, cruise phase is not modelled)

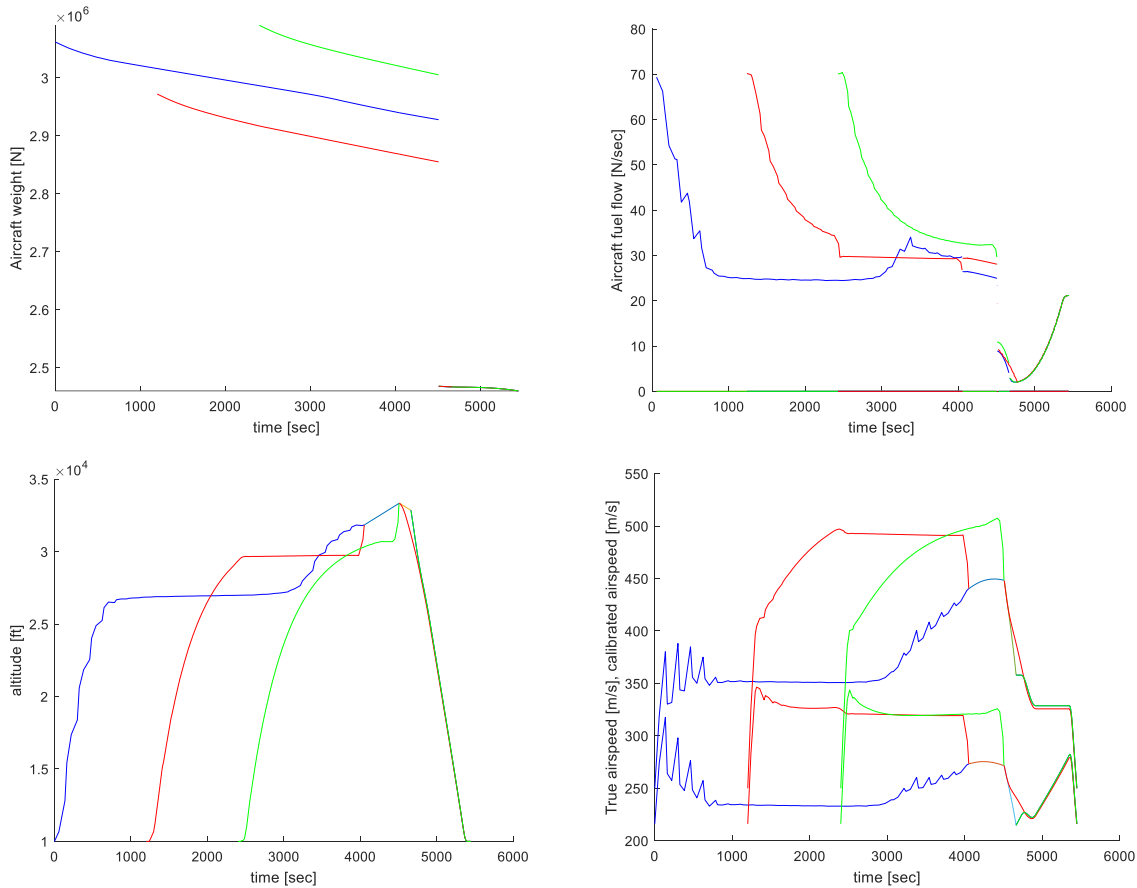


Figure 27: Experiment 5 with formations: a. Aircraft weight vs. time (top left), b. fuel flow vs. time (top right), c. altitude vs. time (bottom left), d. airspeed vs. time (bottom right). Cruise phase is not modelled.

All flights show a deviation in terms of fuel usage which is no larger than 0.5% individually, while the total deviation is even more limited at just 0.12%. These results demonstrate that using the modified Bréguet range equations is a viable alternative for the cruise formation phase, since in terms of accuracy this approach yields desired results. The slight increase in net fuel burn could be explained by the limitation that with these equations, mach number and altitude are kept constant. Therefore, the designed implementation as discussed in Section 6.5 is also verified.

In terms of trajectory, some differences can be noticed between Experiments 4 and 5. Flight KL 641 does not take a detour in its horizontal trajectory anymore to ‘wait’ for the other aircraft. Instead, it just joins flight AF 6 at a later stage. Connected to this phenomenon is the observation that KL 641 keeps a slightly higher velocity while waiting for AF 6, at a slightly higher altitude.

Due to the fact that time is not a part of the Bréguet range equations, it’s not possible to incorporate flight time into the results. The trajectory diagrams therefore show a ‘gap’ for the cruise formation phase, and the time vectors are therefore also not representative in terms of absolute value. Because of this, flight time values between Experiment 4 and 5 cannot be compared. Another possible limitation of this approach is that a wind model cannot be implemented.

## 8 Discussion

### 8.1 Overall evaluation of formation flight experiments

Although the results of the formation flying trajectory optimization experiments are already discussed per experiment in Chapter 7, several more general observations are also made. First of all, between discrete phase switching points, aircraft follow a great circle arc horizontal trajectory. This is to be expected, as no wind effects or climate model is implemented. Only in Experiment 4, flight KL 641 makes a detour to accommodate the timing of the formation rendezvous. Next, it is observed in all experiments except Experiment 5 that as the aircraft become lighter, the altitude increases and the airspeed decreases. This is in line with the drag equation (Eq. 6.10) presented in Section 6.1, since the drag decreases at lower air densities  $\rho$  as well as lower velocities  $V$ . As the lift requirement decreases due to a lighter aircraft weight, it thus becomes more efficient to climb (at higher altitudes the air density decreases) as well as slow down. Experiment 5 assumes a constant altitude and velocity during the cruise phase, so this behavior is not observed here. In these same experiments, it is found that compared to their respective solo trajectories, formation flight velocities are lower. This can be explained by the fact that with lower induced drag levels, form drag becomes more dominant (See Eq. 6.12). This type of drag is more affected by a higher airspeed, so a lower velocity becomes more optimal. At the same time, the optimal altitude in formation flight as determined by the model is actually higher.

In all experiments, jagged lines are visible in the altitude vs. time and the airspeed vs. time graphs. Normally, one would expect a smoother trajectory. The phenomenon observed here is that the model decides to exchange altitude and velocity between each collocation point, since there is no fuel penalty attached to this. The relatively low resolution of 25 intervals per phase and 2 nodes per interval (thus a linear fit) makes these jagged lines more prominent. The optimization is however not significantly negatively affected by this, however in real life the trajectories to be flown would have to be much smoother, especially during climb. At the cost of computation time, a smoother trajectory can be obtained by increasing the number of nodes per interval to 3 or 4 (enabling a higher order approximation) as well as the number of intervals.

In the diagrams for fuel flow, discrete drops and increases in fuel flow can be observed every time an aircraft experiences a change in induced drag reduction factor. This behavior is as expected. Also, small gaps appear at every discrete phase switch in these diagrams. This is because the fuel flow is computed as a numerical derivative from the obtained weight data, meaning that for every phase one less data point is available.

## 8.2 Evaluation of the Hybrid Optimal Control approach

### 8.2.1 Performance

The developed hybrid optimal control approach has demonstrated clear benefits in achieving accurate trajectories with manageable computational effort. The combinatorial problem introduced by making the phase switching structure a part of the optimization is dealt with in an efficient manner by using a branch-and-bound algorithm implementation. Branch-and-bound has especially useful characteristics when considering that the problem complexity of nodes in the solution tree increases with every deeper level of this tree, since more phases are added with more collocation points. The top-down approach where the number of phases gradually increases as branches are cut off ensures that the number of evaluated tree nodes in the more complex lower levels is relatively limited.

Even though the branch-and-bound algorithm already reduced the number of evaluated nodes significantly for three-aircraft experiments such as Experiment 1, the four-aircraft scenario of Experiment 2 showed the true potential of this method. The number of evaluated nodes was limited to just 35 of the possible 347. Without branch-and-bound, such a trajectory optimization problem could not have been solved with the currently used optimal control paradigm. Still, the total computation time for this case was quite high on a quadcore i7-870 desktop, taking over 5 hours to compute.

The associated guess adaptation algorithm, which transforms solutions from parent nodes into initial guesses for their children nodes, has demonstrated to yield initial guesses which in the majority of the cases lead to quick convergence of the optimization. The accuracy of this algorithm also increases further down the solution tree, as the difference between parent and children nodes become smaller. This is especially useful since the complexity of the trajectories at lower tree levels is higher.

### 8.2.2 Reliability

In general, the developed tool performs according to the parameters set without issues. However, it is noted that the reliability of the existing SNOPT solving algorithm gave problems with obtaining the right result. Occasionally, SNOPT would encounter numerical errors while evaluating a node and terminate the optimization. The reason for these errors are unclear, since more often than not a second attempt of the same node evaluation with the same input would solve without problems. Normally, during a non-hybrid optimal control problem evaluation with only one phase structure, this would not pose a serious issue. The automatized nature of the branch-and-bound algorithm however requires a correct solution for every evaluated node because the proceedings of the entire optimization could depend on it. Especially in the case where a false solution has a more desirable objective value, subsequent results could be considered invalid.

Additionally, there is also a significant variation observed in computation times per node. With a good initial guess, the problem should converge quickly to firstly a feasible and then also an optimal solution. Since obtaining accurate initial guesses is not straightforward, it occurs regularly that the initial guess generated by the guess adaptation algorithm is not close enough to the desired feasible solution region to achieve quick convergence. SNOPT then starts ‘searching around’ for a long period. This phenomenon occurred especially in the first



three levels of the solution tree, since here there is a large difference between the trajectories of the node and its parent.

### **8.2.3 Applicability outside formation flying**

Initially, the goal of this thesis project was to investigate hybrid optimal control theory and successfully demonstrate its applicability to aviation trajectory optimization problems. The specific application towards formation flying was chosen later as a focus area, and this implementation has been demonstrated successfully. Still, it is of interest to see whether the developed algorithm could be used in other aviation applications as well. The concept of branch-and-bound for hybrid optimal control is flexible and general enough to be useful in many hybrid optimization problems, as has also been demonstrated in previous research [16], [18]. However, one can conclude that the majority of the development effort in this thesis has been spent on the specific formation implementation with the development of a formation flying phase structure notation and the adaptation of initial guesses between parent and children nodes in the branch-and-bound search tree. The development of the branch-and-bound algorithm itself was relatively straightforward. Therefore, it is noted that the developed algorithms are too specific to formation flying to be used in other types of aircraft trajectory optimization problems without major modification. The general approach however is very promising, so using the approach presented in this thesis to deal with other types of aircraft trajectory optimization problems can be recommended.

### **8.2.4 Limitations**

Although successful results have been obtained with the designed hybrid optimal control approach, several limitations of this method are identified which must be taken into account when using this approach for further experiments. First of all, it is of importance that the relative positioning between aircraft in a formation phase is a discrete aspect of the trajectory which is not taken into account for the model. The aircraft positions are thus chosen arbitrarily, which could lead to a suboptimal result. There is also an uncertainty present in the branch cut off parameters, so a poor choice of the cutoff margin could yield false results or an impractical computational time. Moreover, the optimization model has been kept simple by omitting wind and atmospheric effects. A more elaborate wind and atmospheric model implementation could be necessary to achieve the desired trajectory accuracy. This could however have drastic effects for the computation time. Lastly, it is noted that in its current state, the approach does not fully support a five-aircraft case. Not only will the solution tree be of impractical size with five aircraft, several modifications would also have to be made to the trajectory modelling to enable this.



## 9 Conclusion and Recommendations

### 9.1 Conclusion

#### 9.1.1 Introduction

In this thesis, a Hybrid Optimal Control (HOC) framework which can be used in conjunction with existing optimal control software (GPOPS) is presented. By using hybrid optimal control theory, trajectory optimization problems for systems that are both *discrete time and continuous time* from a mathematical perspective can be solved. While one main objective is to successfully build this algorithm, a case study on multi-aircraft formation flying for civil aviation has been performed which can demonstrate the capabilities of the designed method. Whilst previous research has dealt with either high accuracy trajectories for small formation flying problems or low accuracy modelling of very large formation flying trajectories, in this thesis HOC is used to achieve a high accuracy optimal trajectory for formations of three aircraft and larger, creating a step change in the state of the art with respect to these existing studies.

It can be concluded that the main objective has been achieved. An extension to GPOPS which deals with the hybrid nature of multi-aircraft formation flying optimization problems by exploiting a branch-and-bound algorithm has been developed. Both a three- and a four-aircraft formation optimization case study have been performed to test this developed tool. From the analysis of these case studies the conclusion can be drawn that the tool works as designed and provides a more efficient way to deal with the combinatorial complexity of these problems.

The main objective is also supported by several sub-goals. All sub-goals have been treated in this thesis and can be marked as achieved. Existing literature on the topic of (hybrid) optimal control theory, trajectory optimization and nonlinear programming problems has been researched to obtain adequate knowledge about the topic of this research. Then, an analysis was made of several existing approaches to dealing with HOC problems in aviation trajectory optimization problems. Based on this analysis, a suitable approach to HOC, in the form of a branch-and-bound algorithm, was selected for the development of the framework. Literature on formation flying in civil aviation, including previous research on trajectory optimization in this field, was also researched to determine the requirements of the hybrid optimal control tool for this application.

#### 9.1.2 Model development

The following step was then to get familiar with the working principles behind the existing optimal control software GPOPS and the associated aircraft optimization package FORT. It has been identified which elements needed rewriting or other modifications to enable support for a hybrid optimal control formation flying problem with three aircraft or more. The structure of FORT has been modified to a flexible version which automatically adapts the problem for a given structure of flight phases. After this, the HOC extension could be built by

adding an external branch-and-bound algorithm which evaluates a tree of the solution space top down, with each level adding extra complexity in the phase structure. The algorithm is capable of ‘chopping tree branches’ by evaluating at each level which branches are showing promising improvements to the objective value and which ones are stagnating.

Since the HOC approach focuses on an application for a multi-aircraft formation flying trajectory optimization, a modelling approach for the aircraft dynamics also had to be chosen. Based on the existing models available in FORT, a reduced-order point mass system was used, with the state consisting of  $7 + N$  variables, where  $N$  is the number of aircraft in formation. For control, three variables were used. From this point-mass system model the change in weight of each aircraft, which equals the fuel flow, is derived. The total objective value is then determined by the sum of all aircraft initial weights  $W_0^i$  minus all aircraft final weights  $W_f^i$ . Relevant time constraints, state limit constraints and path constraints were applied, and finally an assessment of the chosen assumptions and simplifications for the model was also made.

### 9.1.3 Formation flight case studies

To test the developed algorithm, a case with a three-aircraft formation flight was evaluated and compared with the results of a full enumeration of the solution space. Then, after validating the correct functionality of the algorithm, a significantly more complex four-aircraft formation scenario with two different induced drag reduction factor schemes was evaluated. Next to this, another three-aircraft scenario case study was performed which focuses on the existing flight schedule of KLM, Air France and Virgin Atlantic. From the results of these optimization case studies, it can be concluded that three- and four-aircraft transatlantic formation flights demonstrate significant fuel benefit when compared to their respective solo trajectories. This is at the expense of additional total flight time, but generally the added flight times are relatively limited and should not pose a significant planning or operating costs disadvantage. Still, it is of importance to consider this potential drawback when planning real-life aircraft trajectories.

A demonstrative scenario was also performed in which the aircraft dynamics for the largest formation cruise phase were not modelled with a point-mass system, but rather represented using modified Bréguet range equations for formation flight. The same flights were taken as in the presented three-aircraft scenario based on existing flight schedules, for comparison. A general conclusion is that this method shows potential for simplifying the optimization model, as only slight deviations were observed from the results of the regular experiment. However, computational time did not decrease, but actually increased using this model as GPOPS had troubles with convergence using this method. Therefore, in this thesis this method did not yield any additional benefit.

Regarding the modelling approach, it was chosen not to consider the order of the aircraft in the formation as an optimization variable. However, this aspect could significantly influence the effectiveness of the optimization, since generally heavier aircraft benefit more from being in a trailing position. Therefore, this can be identified as a potential suboptimality for the results. Secondly, the resolution of the solution might not be sufficient, since jagged

lines were present in some trajectories which could have been eliminated by using more collocation points. This low resolution however does not significantly influence the actual trajectory performance characteristics. Lastly, it is identified that several simplifications are made to the model, such as the lack of wind and realistic atmospheric data, which might influence the accuracy of the results significantly.

#### **9.1.4 Effectiveness of the Hybrid Optimal Control approach**

Based on the performed case studies, several conclusions can be drawn about the effectiveness of the developed HOC approach for multi-aircraft formation flying trajectory optimization. First of all, the transformation of FORT towards a flexible set-up where the phase structure can be easily changed has been successful, as it enabled the development of smart HOC techniques. The chosen approach to solve the hybrid trajectory optimization problem efficiently, using a branch-and-bound algorithm, has proved to be effective in reducing the number of discrete evaluations required, especially in a four-aircraft scenario where the number of discrete formation possibilities exceeds 300. Using multi-threading also enabled a more efficient evaluation of the discrete solution tree. A great challenge has been to develop and optimize the initial guess adaptation tool, which converts initial guesses to the correct format for a different phase structure. The adapted guess accuracy heavily influences the convergence behavior of GPOPS, and thus also the computation time. In the end however, the guess adaptation tool proved accurate enough in most cases.

With the chosen branch-and-bound method, attention had to be paid to the correctness of the lower bounds of each node, since the cut-off conditions are determined heuristically. In several cases, tree branches were kept unnecessarily because the differences between solutions were small. However, the opposite, where a false cut-off occurs, is a possible risk as well. Also, even though the number of evaluated nodes was reduced drastically using branch-and-bound, computation times were still suboptimal, with the four-aircraft case requiring more than 5 hours to reach an optimum. Therefore, it can be concluded that branch-and-bound alone might not be sufficient to enable such trajectory optimizations for practical use, and that further improvements to the optimization model might be necessary.

## **9.2 Recommendations**

It has been demonstrated that the objectives for the hybrid optimal control approach have largely been met. However, based on the conclusions drawn in this chapter, several recommendations can be made for future research to extend or improve upon the work presented in this thesis.

Firstly, the hybrid optimal control approach can be further developed. For example, it can be investigated if computation times can be reduced significantly by removing tree nodes and branches containing phase structures which can be identified a priori as suboptimal. Another related aspect which can be investigated further is improving the branch cut-off conditions, since this heavily influences the computational efficiency as well as the reliability.

In terms of the trajectory modelling, several directions could be taken which might further improve the efficiency for the optimization in GPOPS. To start, it is recognized that the discrete combinatorial problem of determining the correct phase structure depends mostly on horizontal trajectories. Based on this observation, a method could be developed in which only these 2D horizontal trajectories are evaluated during the branch-and-bound run. Finally, the found optimal phase structure can then be evaluated in 4D, resulting in a complete trajectory. Using this approach, it might also be possible to achieve higher resolution trajectories using more collocation points. Another direction which could be taken is to further investigate the use of Bréguet range equations for cruise phases, which would reduce the complexity of the model significantly.

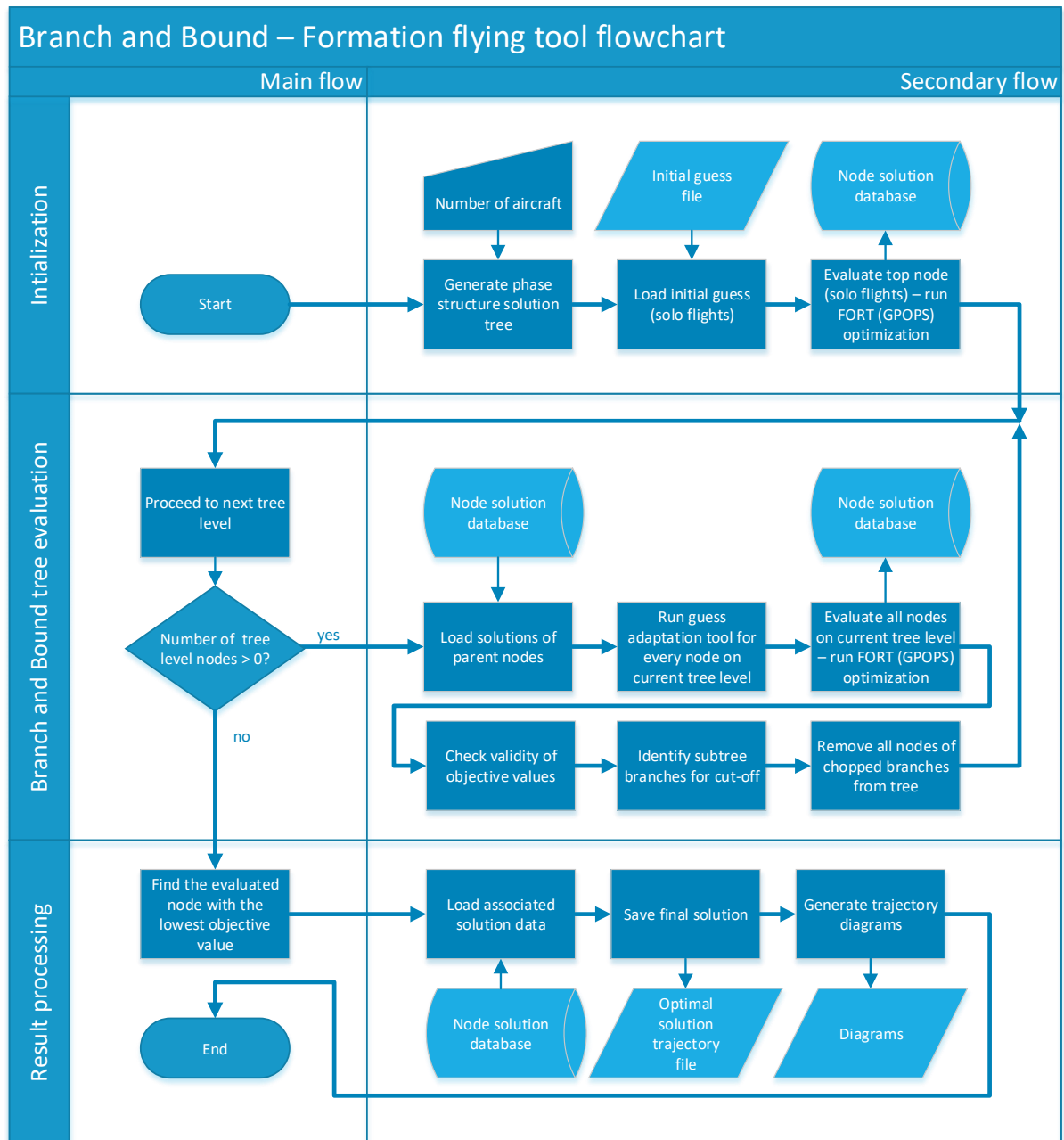
To make the simulations more elaborate and accurate, there are also multiple identified research continuations. First of all, in this study only one aircraft type was evaluated, the Boeing 747-400. It is therefore of interest to determine how the optimization behaves with different aircraft models. The interaction between multiple aircraft types in one formation could be investigated, since there might be a variety in the induced drag reduction behavior when different aircraft types are involved. Secondly, an implementation of a more accurate atmosphere model, as well as a wind model, could drastically improve the realism of the obtained results, at the expense of a more complicated model. Thirdly, including the formation aircraft order into the optimization could yield some further fuel burn savings, since it would ensure that the right aircraft will lead or trail in the formation. Another factor which would improve the realism is the inclusion of a minimum initial fuel load which forces each aircraft to take enough fuel to fly the trajectories solo, since the current approach where just the minimum fuel is taken would pose serious risks in practical use. Lastly, a more elaborate cost function could be developed where not only fuel use, but also other aspects related to the Direct Operating Cost (DOC), such as flight time, are included.

Finally, as demonstrated in previous literature, the methodology used in this research could be suitable for various other research topics in the field of aircraft trajectory optimization. Many discrete phenomena are identified in problems concerning aircraft trajectory planning, and using a hybrid optimal control approach for solving the discrete aspects as well as the continuous aspects of a planning problem has proven to yield useful results.

### 9.3 Final remarks

To conclude, it can be observed that the usefulness of a HOC approach is very high for the chosen application of multi-aircraft formation flying. Although the applied method is still not completely optimal in terms of practicality and accuracy, the results that have been found are very promising and could significantly improve the operations of civil aviation. This thesis confirms that investigating hybrid optimal control for this application has been a valuable exercise, and that there are many new opportunities which could be explored as a continuation of the research presented here. Multi-aircraft formation flying shows promising fuel saving potential. Therefore, given the number of flights that are performed every day, the impact of saving fuel just by optimizing their trajectories using formation flying can be very high on a global scale, both environmentally and economically.

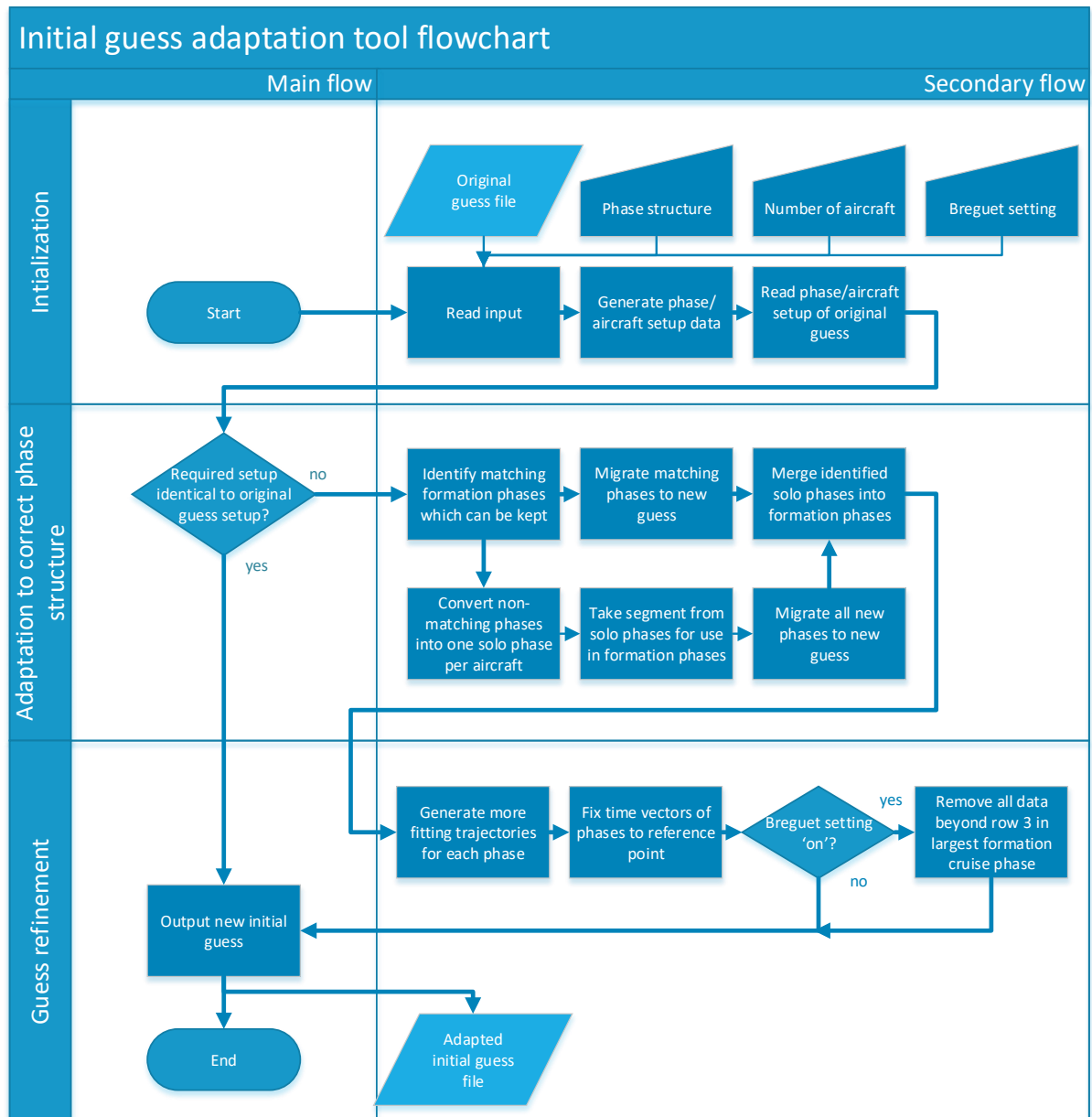
# A Branch-and-bound algorithm flowchart





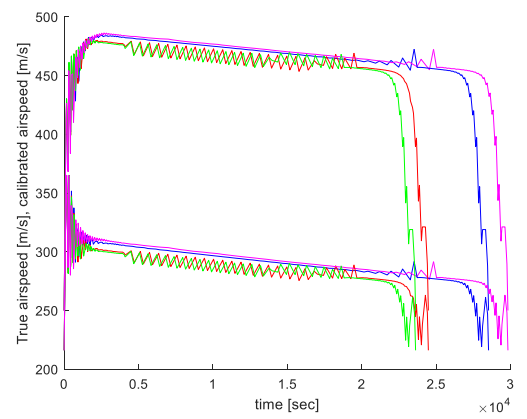
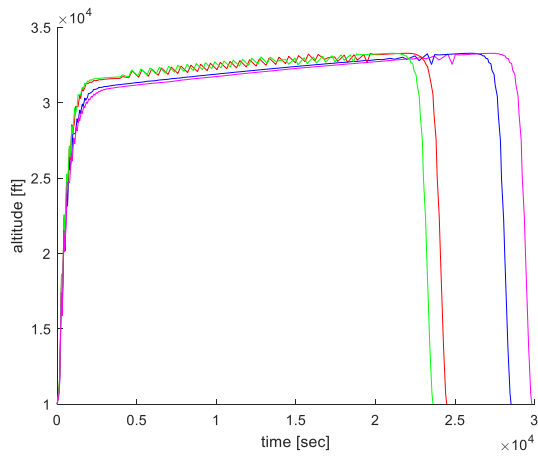
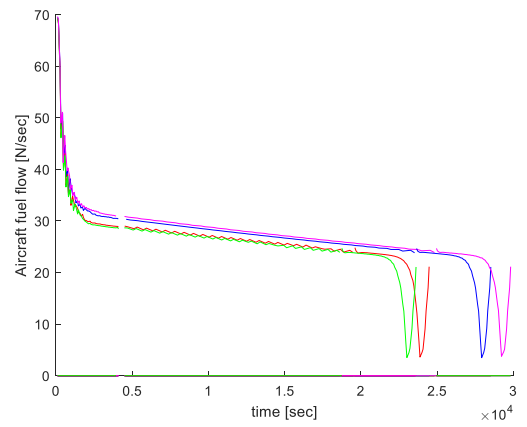
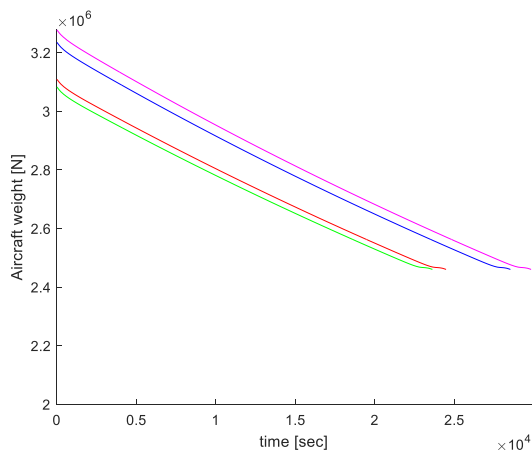
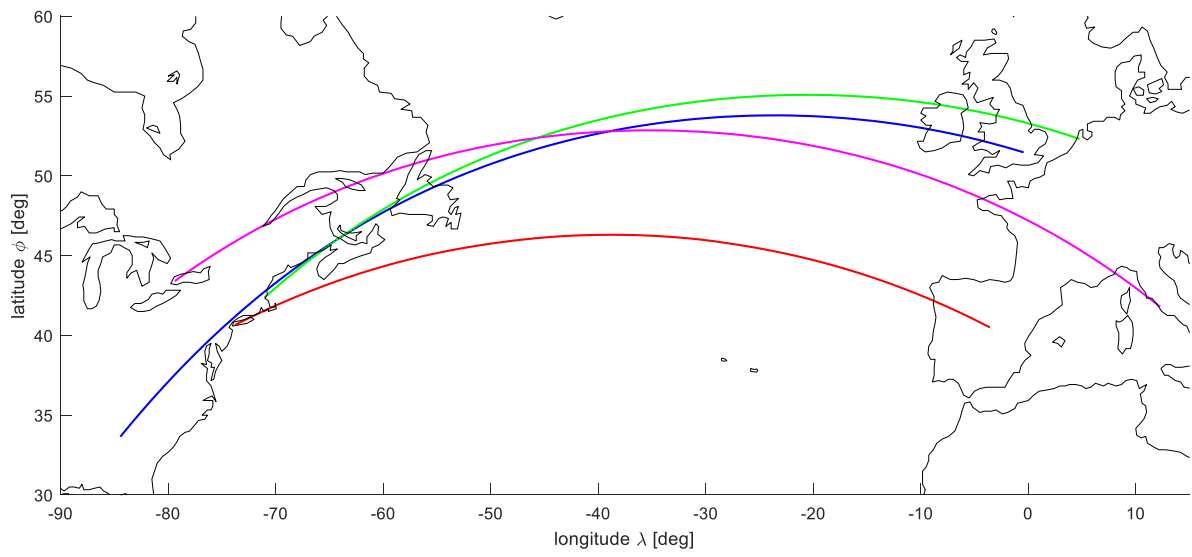


## B Initial guess adaptation tool flowchart



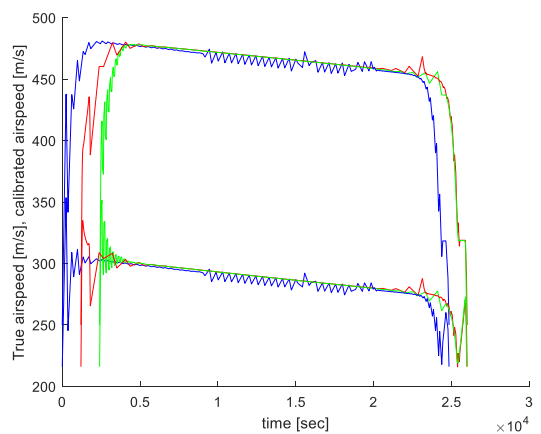
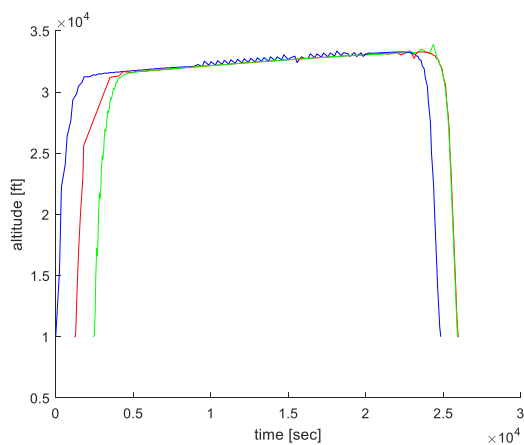
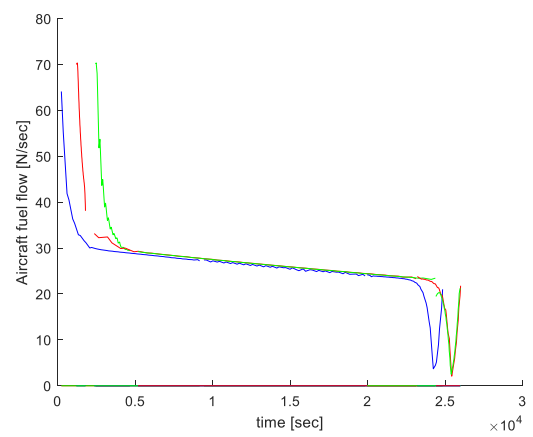
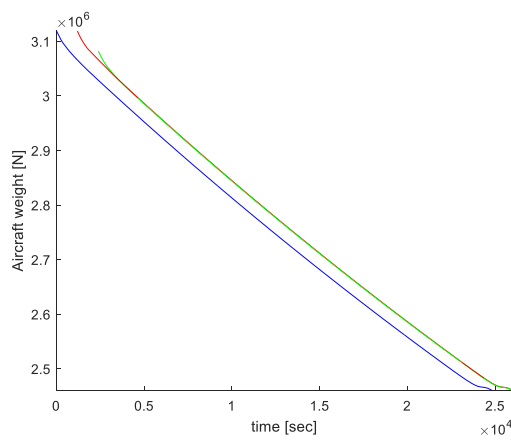
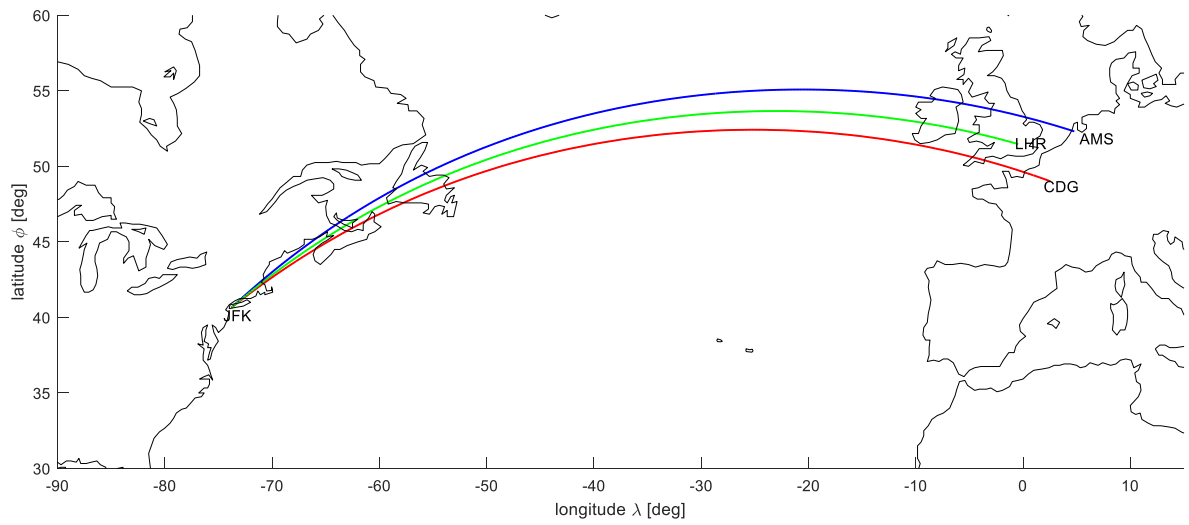


## C Solo flight trajectories (Experiment 1,2,3)





## D Solo flight trajectories (Experiment 4,5)





## References

- [1] SESAR Consortium, “European ATM Master Plan Executive View,” 2015.
- [2] M. Buss, O. Von Stryk, R. Bulirsch, and G. Schmidt, “Towards Hybrid Optimal Control,” *Automatisierungstechnik*, vol. 48, no. 9\_2000, p. 448, 2000.
- [3] M. E. G. van Hellenberg Hubar, “Multiple-Phase Trajectory Optimization for Formation Flight in Civil Aviation,” Delft University of Technology, 2016.
- [4] A. V Rao, C. L. Darby, and M. Patterson, “User ’ s Manual for GPOPS Version 3 . 3 : A MATLAB Software for Solving Optimal Control Problems Using Pseudospectral Methods,” *Control*, no. December, 2010.
- [5] H. G. Visser, *Aircraft Performance Optimization - Reader*, vol. 9, no. 2. Delft: TU Delft Faculty of Aerospace Engineering, 2014.
- [6] J. T. Betts, “Survey of Numerical Methods for Trajectory Optimization,” *J. Guid. Control. Dyn.*, vol. 21, no. 2, pp. 193–207, 1998.
- [7] P. B. S. LISSAMAN and P. SHOLLENBERGER, CARL A. (California Institute of Technology, “Formation Flight of Birds,” *Science (80-. )*, vol. 168, pp. 1003–1005, 1970.
- [8] S. Hartjes, “An Optimal Control Approach to Helicopter Noise and Emissions Abatement,” TU Delft, 2015.
- [9] L. S. Pontryagin, *Mathematical Theory of Optimal Processes*. CRC Press, 1962.
- [10] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2010.
- [11] O. von Stryk and R. Bulirsch, “Direct and indirect methods for trajectory optimization,” *Ann. Oper. Res.*, vol. 37, no. 1, pp. 357–373, 1992.
- [12] M. Branicky, V. Borkar, and S. K. Mitter, “A unified framework for hybrid control: Model and optimal control theory,” *IEEE Trans. Automat. Contr.*, vol. 43, no. 1, pp. 31–45, 1998.
- [13] M. Kamgarpour, “Optimal Control of Hybrid Systems in Air Traffic Applications,” University of California, Berkeley, 2011.
- [14] FAA, “NextGen.” [Online]. Available: <https://www.faa.gov/nextgen/>. [Accessed: 10-Feb-2017].
- [15] M. Soler, A. Olivares, and E. Staffetti, “Hybrid Optimal Control Approach to Commercial Aircraft Trajectory Planning,” *J. Guid. Control. Dyn.*, vol. 33, no. 3, pp. 985–991, May 2010.
- [16] P. Bonami, A. Olivares, M. Soler, and E. Staffetti, “Multiphase Mixed-Integer Optimal

- Control Approach to Aircraft Trajectory Optimization,” *J. Guid. Control. Dyn.*, vol. 36, no. 5, pp. 1267–1277, 2013.
- [17] O. Von Stryk and M. Glocker, “Decomposition of Mixed-Integer Optimal Control Problems Using Branch and Bound and Sparse Direct Collocation,” *Proc. ADPM 2000 -- 4th Int. Conf. Autom. Mix. Process. Hybrid Dyn. Syst.*, vol. 19, pp. 99–104, 2000.
- [18] M. Soler, B. Zou, and M. Hansen, “Flight trajectory design in the presence of contrails: Application of a multiphase mixed-integer optimal control approach,” *Transp. Res. Part C Emerg. Technol.*, vol. 48, pp. 172–194, Nov. 2014.
- [19] P. Bonami, M. Kiliç, and J. Linderoth, “Algorithms and software for convex mixed integer nonlinear programs,” *Mix. Integer Nonlinear Program.*, pp. 1–39, 2012.
- [20] P. Bonami *et al.*, “An algorithmic framework for convex mixed integer nonlinear programs,” *Discret. Optim.*, vol. 5, no. 2, pp. 186–204, May 2008.
- [21] M. Soler, M. Kamgarpour, J. Lloret, and J. Lygeros, “A Hybrid Optimal Control Approach to Fuel-Efficient Aircraft Conflict Avoidance,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1826–1838, 2016.
- [22] M. Kamgarpour, M. Soler, C. J. Tomlin, A. Olivares, and J. Lygeros, *Hybrid optimal control for aircraft trajectory design with a variable sequence of modes*, vol. 18, no. PART 1. IFAC, 2011.
- [23] X. Xu and P. J. Antsaklis, “Optimal control of switched systems based on parameterization of the switching instants,” *IEEE Trans Autom Control* 49, 2004.
- [24] S. A. Ning, “Aircraft Drag Reduction through Extended Formation Flight,” 2011.
- [25] W. B. Blake, S. R. Bieniawski, and T. C. Flanzer, “Surfing aircraft vortices for energy,” *J. Def. Model. Simul. Appl. Methodol. Technol.*, vol. 12, no. 1, pp. 31–39, 2015.
- [26] W. Blake and D. Multhopp, “Design, performance and modeling considerations for close formation flight,” *23rd Atmos. Flight Mech. Conf.*, vol. 2, no. AIAA-98-4343, pp. 476–486, 1998.
- [27] S. A. Ning, T. C. Flanzer, and I. M. Kroo, “Aerodynamic Performance of Extended Formation Flight,” *J. Aircr.*, vol. 48, no. 3, pp. 855–865, 2011.
- [28] H. G. Visser, B. F. Santos, and C. M. A. Verhagen, “A Decentralized Approach to Formation Flight Routing of Long-Haul Commercial Flights,” pp. 1–34.
- [29] H. G. Visser and S. Hartjes, “Economic and environmental optimization of flight trajectories connecting a city-pair,” *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.*, vol. 228, no. 6, pp. 980–993, 2014.
- [30] D. Garg *et al.*, “Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method,” *Comput. Optim. Appl.*, vol. 49, no. 2, pp. 335–358, 2011.
- [31] M. Voskuijl, “Cruise range in formation flight,” Delft, NL, 2016.



- [32] EUROCONTROL, “European ATM Master Plan 2015,” p. 140, 2015.
- [33] “Virgin Atlantic Sells Stake To Air France-KLM, Strengthens Alliance With Delta Air Lines,” *Forbes*. [Online]. Available: <https://www.forbes.com/sites/grantmartin/2017/08/01/virgin-atlantic-sells-stake-to-air-france-klm-strengthens-alliance-with-delta-air-lines/#3571744078e6>. [Accessed: 11-Oct-2017].
- [34] “Flight Schedules | Flight Timetable - Wego.com.” [Online]. Available: <https://www.wego.com/schedules>. [Accessed: 11-Oct-2017].