UNIVERSIDADE
BEIRA INTERIOR

4D Commercial Trajectory Optimization for Fuel Saving and Environmental Impact Reduction

Kawser Ahmed

Tese para obtenção do Grau de Doutor em
Engenharia Aeronáutica
(3º ciclo de estudos)

Orientador: Prof. Doutor Kouamana Bousson

Fevereiro de 2022

Date of Doctoral Proofs

02 February 2022

Jury Panel

President of the Jury: Professor Dr. Joaquim Mateus Paulo Serra
Full Professor at the University of Beira Interior

Examiner: Professor Dr. Jorge Manuel Martins Barata
Full Professor at the University of Beira Interior

Examiner: Professor Dr. José António Menezes Felippe de Souza
Associate Professor at the University of Beira Interior

Supervisor: Professor Dr. Kouamana Bousson
Associate Professor at the University of Beira Interior

Examiner: Professor Dr. Francisco Miguel Ribeiro Proença Brójo
Assistant Professor at the University of Beira Interior

Principal Examiner: Dr. Elena Nikolaevna Koroleva Duarte
Assistant Researcher at the Faculty of Sciences of the University of Lisbon

Examiner: Professor Dr. André Resende Rodrigues da Silva
Assistant Professor at the University of Beira Interior

Principal Examiner: Dr. Sandra Clara Rodrigues Antunes
Specialist at OGMA – Aeronautical Industry of Portugal

Acknowledgments

Firstly, I would like to express my sincerest appreciation to my mentor and PhD supervisor, Professor Kouamana Bousson, who has endlessly supported and guided me throughout this thesis, without which this work would not have been possible. I will be forever grateful for your kindness.

I would also like to thank my parents and sister for their continuous support and patience, which has made it possible for me to be here. Also, thanks to all my friends, for always being there and for their understanding and support.

I would like to give special thanks to the Aeronautics and Astronautics Research Center (AeroG) of the Associate Laboratory for Energy, Transports and Aeronautics (LAETA) for having supported me and provided all the necessary material to carry out my PhD thesis, as well as for having provided me with an excellent working environment for research.

I would like to thank Santander Universidades (the BID/ICI-FE/Santander Universidades-UBI/ 2016) for the grant that supported partially my PhD work from October 2016 to August 2017 and the Portuguese National Funding Agency for Science, Research and Technology (FCT - Fundação para a Ciência e a Tecnologia) for supporting me through the grant (Ref - SFRH/BD/140577/2018) from October 2018 until the end of my PhD research work in March 2021.

Finally, I would like to give thanks to all the professors and colleagues of the Department of Aerospace Sciences of the University of Beira Interior (UBI), especially Milca Coelho for helping me with productive discussions, invaluable help, and friendship.


FCT
Fundação
para a Ciência
e a Tecnologia

REPÚBLICA
PORTUGUESA
CIÊNCIA, TECNOLOGIA
E ENSINO SUPERIOR

FSE
60
ANOS
Investir nas pessoas
Fundo Social Europeu


Abstract

The main purpose of the thesis is to optimize commercial aircraft 4D trajectories to improve flight efficiency and reduce fuel consumption and environmental impact caused by airliners. The Trajectory Optimization Problem (TOP) technique can be used to accomplish this goal. The formulation of the aircraft TOP involves the mathematical model of the system (i.e., dynamics model, performance model, and emissions model of the aircraft), Performance Index (PI), and boundary and path constraints of the system. Typically, the TOP is solved by a wide range of numerical approaches. They can be classified into three basic classes of numerical methods: indirect methods, direct methods, and dynamic programming. In this thesis, several instances of problems were considered to optimize commercial aircraft trajectories.

Firstly, the problem of optimal trajectory generation from predefined 4D waypoint networks was considered. A single source shortest path algorithm (Dijkstra's algorithm) was applied to generate the optimal aircraft trajectories that minimize aircraft fuel burn and total trip time between the initial and final waypoint in the networks. Dijkstra's Algorithm (DA) successfully found the path (trajectory) with the lowest cost (i.e., fuel consumption, and total trip time) from the predefined 4D waypoint networks.

Next, the problem of generating minimum length optimal trajectory along a set of predefined 4D waypoints was considered. A cubic spline parameterization was used to solve the TOP. The state vector, its time derivative, and control vector are parameterized using Cubic Spline Interpolation (CSI). Consequently, the objective function and constraints are expressed as functions of the value of state and control at the temporal nodes, this representation transforms the TOP into a Nonlinear Programming (NLP) problem, which is then solved numerically using a well-established NLP solver. The proposed method generated a smooth 4D optimal trajectory with very accurate results.

Following, the problem considers generating optimal trajectories between two 4D waypoints. Dynamic Programming (DP) a well-established numerical method was considered to solve this problem. The traditional DP bears some shortcomings that prevent its use in many practical real-time implementations. This thesis proposes a Modified Dynamic Programming (MDP) approach which reduces the computational effort and overcomes the drawbacks of the traditional DP. The proposed MDP approach was successfully implemented to generate optimal trajectories that minimize aircraft fuel

consumption and emissions in several case studies, the obtained optimal trajectories are then compared with the corresponding reference commercial flight trajectory for the same route in order to quantify the potential benefit of reduction of aircraft fuel consumption and emissions. The numerical examples demonstrate that the MDP can successfully generate fuel and emissions optimal trajectory with little computational effort, which implies it can also be applied to online trajectory generation.

Finally, the problem of predicting the fuel flow rate from actual flight data or manual data was considered. The Radial Basis Function (RBF) neural network was applied to predict the fuel flow rate in the climb, cruise, and descent phases of flight. In the RBF neural network, the true airspeed and flight altitude were taken as the input parameters and the fuel flow rate as the output parameter. The RBF neural network produced a highly accurate fuel flow rate model with a high value of coefficients of determination, together with the low relative approximation errors. Later on, the resulted fuel flow rate model was used to solve a 4D TOP by optimizing aircraft green cost between two 4D waypoints.

Keywords

4D trajectory; trajectory optimization; base of aircraft data; Dijkstra's algorithm; spline parameterization; nonlinear programming; dynamic programming; fuel-efficient; aircraft emissions; RBF neural network; fuel flow rate prediction.

Resumo

O principal objetivo desta tese é otimizar as trajetórias em 4D de aeronaves comerciais, de forma a melhorar a eficiência de voo e reduzir o consumo de combustível e o impacto ambiental causado pelos aviões. A técnica de otimização de trajetória pode ser utilizada para atingir este objetivo. A formulação do problema de otimização de trajetória de uma aeronave envolve o modelo matemático do sistema (isto é, modelo de dinâmica, modelo de desempenho, e modelo de emissões de aeronaves), a função objetiva e os limites e restrições do sistema. Normalmente, o problema de otimização de trajetória é solucionado por uma ampla variedade de abordagens numéricas, que podem ser classificadas em três classes básicas de métodos numéricos: métodos indiretos, métodos diretos e programação dinâmica. Nesta tese, foram consideradas várias instâncias de problemas para otimizar trajetórias de aeronaves comerciais.

Em primeiro lugar, foi considerado um problema de geração de trajetória ótima em 4D a partir de redes de waypoints predefinidas. Para tal, foi aplicado um algoritmo de single source shortest path (neste caso, algoritmo de Dijkstra), de forma a gerar trajetórias ótimas que minimizem o consumo de combustível da aeronave e o seu tempo total de viagem. O algoritmo de Dijkstra encontrou com sucesso a trajetória com menor custo, isto é, a trajetória de menor consumo de combustível e menor tempo total de viagem, a partir da rede predefinida de waypoints.

Em seguida, foi considerado o problema de gerar uma trajetória ótima em 4D de comprimento mínimo ao longo de um conjunto de waypoints predefinidos. Para tal, foi utilizada uma parametrização da spline cúbica. O vetor de estado, a sua derivada e o vetor de controlo são parametrizados utilizando a interpolação cúbica da spline. Consequentemente, a função objetivo e as restrições são expressas como funções do valor de estado e controlo nos nós temporais. Esta representação transforma o problema de otimização de trajetória em um problema de programação não-linear, que por sua vez, é resolvido numericamente por um solucionador já bem estabelecido de programação não-linear. O método proposto gerou uma trajetória ótima em 4D com resultados precisos.

Posteriormente, considerou-se o problema de geração de trajetórias ótimas em 4D entre dois waypoints. Para solucionar este problema foi utilizado a programação dinâmica que é um método numérico já bem estabelecido. A programação dinâmica apresenta algumas deficiências que impedem o seu uso em muitas aplicações práticas de tempo-real. Por isso, esta tese propõe uma abordagem de programação dinâmica modificada que reduz

o esforço computacional e supera as desvantagens do Programação Dinâmica tradicional. A abordagem programação dinâmica modificada proposta, foi implementada com sucesso em vários casos de estudo, em que foram geradas trajetórias ótimas que minimizam o consumo de combustível da aeronave e as suas emissões. Estas trajetórias são, posteriormente, comparadas com a trajetória de voo comercial de referência, para quantificar a potencial redução do consumo de combustível da aeronave e das suas emissões. Os exemplos numéricos demonstram que a programação dinâmica modificada pode gerar com sucesso e com pouco esforço computacional trajetórias ótimas para o combustível e as emissões, o que sugere que este método pode ser aplicado em situações online, isto é, geração de trajetórias online.

Por fim, foi considerado o problema de previsão da taxa temporal de consumo de combustível (FF) a partir de dados de voo reais. A rede neural da função de base radial (RBF) foi aplicada para prever a essa mesma taxa temporal nas fases de voo: subida, cruzeiro e descida. Na aplicação da rede neural RBF, a velocidade real e a altitude de voo foram consideradas como parâmetros de entrada e a FF foi considerada como parâmetro de saída. A rede neural RBF foi capaz de produzir um modelo adequado para estimar corretamente essa taxa temporal, com um elevado valor de coeficientes de determinação, juntamente com baixos valores nos erros relativos de aproximação. Posteriormente, este modelo de FF foi utilizado para resolver o problema de otimização de trajetórias em 4D, em que o custo total entre dois waypoints foi otimizado.

Palavras-chave

Trajetória em 4D; otimização de trajetórias; base de dados de aeronaves; algoritmo de Dijkstra; parametrização da spline; programação não-linear; programação dinâmica; baixo consumo de combustível; emissões da aeronave; rede neural RBF; previsão da taxa temporal de consumo de combustível.

Table of Contents

List of Figures

List of Tables

List of Acronyms

3-DOF	–	Three Degree of Freedom
4D	–	Four-Dimensions
6-DOF	–	Six Degree of Freedom
ACO	–	Ant Colony Algorithms
ANN	–	Artificial Neural Network
ATM	–	Air Traffic Management
BADA	–	Base of Aircraft Data
BFFM2	–	Boeing Fuel Flow Method 2
BVP	–	Boundary Value Problem
CCO	–	Continuous Climb Operation
CDO	–	Continuous Descent Operation
CGL	–	Chebyshev-Gauss-Lobatto
CI	–	Cost Index
CoV	–	Calculus of Variations
CPM	–	Chebyshev Pseudospectral Method
CSI	–	Cubic Spline Interpolation
DA	–	Dijkstra's Algorithm
DI	–	Differential Inclusion
DOC	–	Direct Operation Cost
DP	–	Dynamic Programming
EI	–	Emission Index
EOM	–	Equation of Motion
EU	–	European Union
FFNN	–	Feed-Forward Neural Network
FL	–	Flight Level
FMS	–	Flight Management System
FMC	–	Flight Management Computer
GA	–	Genetic Algorithms
GDOC	–	Green Direct Operating Cost
GMBM	–	Global Market-Based Measure
GPM	–	Gauss Pseudospectral Method
HJB	–	Hamilton-Jacobi-Bellman
HLGL	–	Hermite-Legendre-Gauss-Lobatto

ICAO	–	International Civil Aviation Organization
IDP	–	Iterative Dynamic Programming
IP	–	Interior-Point
ISA	–	International Standard Atmosphere
LG	–	Legendre-Gauss
LGL	–	Legendre-Gauss-Lobatto
LGR	–	Legendre-Gauss-Radau
LMS	–	Least Mean Square
LPM	–	Legendre Pseudospectral Method
MDP	–	Modified Dynamic Programming
MPBVP	–	Multi-point Boundary Value Problem
MS-DP	–	Moving Search space Dynamic Programming
MSL	–	Mean Sea Level
NextGen	–	Next Generation Air Transportation System
NLP	–	Nonlinear Programming
OCP	–	Optimal Control Problem
OPF	–	Operations Performance File
PI	–	Performance Index
PS	–	Pseudospectral
PSO	–	Particle Swarm Algorithms
PMP	–	Pontryagin's Maximum Principle
PO	–	Principle of Optimality
RAE	–	Relative Approximation Error
RBF	–	Radial Basis Function
RPM	–	Radau Pseudospectral Method
SA	–	Simulated Annealing
SESAR	–	Single European Sky ATM Research
SQP	–	Sequential Quadratic Programming
TBO	–	Trajectory-Based Operation
ToC	–	Top of Climb
ToD	–	Top of Descent
TOP	–	Trajectory Optimization Problem
TPBVP	–	Two-point Boundary Value Problem
UAV	–	Unmanned Air Vehicle

List of Symbols

Symbol		Description	Unit
α	–	Angle of attack	$[\mathrm{rad}]$
β_T	–	ISA temperature gradient	$[\mathrm{K/m}]$
γ	–	Flight path angle	$[\mathrm{rad}]$
δ	–	Activation function	N/A
δ_{amb}	–	Pressure ratio	N/A
Δ	–	Defect constraint	N/A
$\Delta\mathrm{T}$	–	Temperature deviation	$[\mathrm{K}]$
ε	–	Tolerance	N/A
η	–	Thrust specific fuel consumption	$\left[\mathrm{kg/(min \cdot kN)}\right]$
θ_{amb}	–	Temperature ratio	N/A
κ	–	Adiabatic index of air	N/A
λ	–	Longitude	$[\mathrm{deg}]$
$\lambda_c(t) \in \mathbb{R}^n$	–	Co-state	N/A
μ	–	Bank angle	$[\mathrm{rad}]$
ρ	–	Air density	$\left[\mathrm{kg/m^3}\right]$
ρ_0	–	Standard atmospheric density	$\left[\mathrm{kg/m^3}\right]$
σ	–	Contraction rate	N/A
ς	–	Number of grid points	N/A
Φ	–	Mayer term of performance index	N/A
φ	–	Latitude	$[\mathrm{deg}]$
ϕ	–	Relative humidity	$[\mathrm{kg/kg}]$
ψ	–	Heading	$[\mathrm{rad}]$
Ψ	–	Boundary constraints	N/A
ω	–	Specific humidity	$[\mathrm{kg/kg}]$
Ω	–	Domain of feasible controls	N/A
a	–	ISA speed of sound	$[\mathrm{m/s}]$

a_0	–	Speed of sound	$[\mathrm{m/s}]$
a	–	Earth semi-major axis	$[\mathrm{m}]$
AE	–	Aircraft emissions	$[\mathrm{g}]$
B	–	Block of grid points	N/A
c_{eq}	–	Equality constraints	N/A
c_{inq}	–	Inequality Constraints	N/A
C	–	Number of quantized control values	N/A
C_{CO_2}	–	Carbon cost coefficient	$[€/\mathrm{kg}]$
C_D	–	Drag coefficient	N/A
C_{D0}	–	Parasite drag coefficient	N/A
C_{D2}	–	Induced drag coefficient	N/A
$C_{\mathrm{f}1}$	–	1st thrust specific fuel consumption coefficients	$[\mathrm{kg/(min \cdot kN)}]$
$C_{\mathrm{f}2}$	–	2nd thrust specific fuel consumption coefficients	$[\mathrm{knots}]$
$C_{\mathrm{f}3}$	–	1st descent fuel flow coefficients	$[\mathrm{kg/min}]$
$C_{\mathrm{f}4}$	–	2nd descent fuel flow coefficients	$[\mathrm{ft}]$
$C_{\mathrm{f\,cr}}$	–	Cruise fuel flow correction coefficient	N/A
C_{FF}	–	Fuel cost coefficient	$[€/\mathrm{kg}]$
C_{gas}	–	Emission gases cost coefficient	$[€/\mathrm{kg}]$
C_L	–	Lift coefficient	N/A
$Cost_{CO_2}$	–	Carbon dioxide emissions cost	$[€]$
$Cost_{FF}$	–	Fuel cost	$[€]$
$Cost_{emi\_gas}$	–	Emission gases cost	$[€]$
$C_{Tc,1}$	–	1st maximum climb thrust coefficients	$[\mathrm{N}]$
$C_{Tc,2}$	–	2nd maximum climb thrust coefficients	$[\mathrm{ft}]$
$C_{Tc,3}$	–	3rd maximum climb thrust coefficients	$[1/\mathrm{ft}^2]$
$C_{Tc,4}$	–	1st thrust temperature coefficient	$[\mathrm{K}]$
$C_{Tc,5}$	–	2nd thrust temperature coefficient	$[1/\mathrm{K}]$

C_{Tcr}	–	Maximum cruise thrust coefficient	N/A
$C_{T\ \mathrm{des,high}}$	–	High-altitude descent thrust coefficient	N/A
$C_{T\ \mathrm{des,low}}$	–	Low-altitude descent thrust coefficient	N/A
CI	–	Cost index	$[\mathrm{kg/\ min}]$
CO		Carbon monoxide	$[\mathrm{g}]$
CO_2	–	Carbon dioxide	$[\mathrm{g}]$
D	–	Drag	$[\mathrm{N}]$
D_{cruise}	–	Cruise drag	$[\mathrm{N}]$
e	–	Eccentricity	N/A
E_{CO}	–	Carbon monoxide emission	$[g]$
$E_{\mathrm{CO_2}}$	–	Carbon dioxide emission	$[g]$
E_{HC}	–	Hydrocarbon emission	$[g]$
$E_{\mathrm{H_20}}$	–	Water vapor emission	$[g]$
$E_{\mathrm{NO_x}}$	–	Oxides of nitrogen emission	$[g]$
$E_{\mathrm{SO_2}}$	–	sulfur dioxide emission	$[g]$
EI	–	Emission index	$[\mathrm{g/\ kg}]$
EI_{CO}	–	Emission index of carbon monoxide	$[\mathrm{g/\ kg}]$
$EI_{\mathrm{CO_2}}$	–	Emission index of carbon dioxide	$[\mathrm{g/\ kg}]$
EI_{HC}	–	Emission index of hydrocarbon	$[\mathrm{g/\ kg}]$
$EI_{\mathrm{H_20}}$	–	Emission index of water vapor	$[\mathrm{g/\ kg}]$
$EI_{\mathrm{NO_x}}$	–	Emission index of oxides of nitrogen	$[\mathrm{g/\ kg}]$
$EI_{\mathrm{SO_2}}$	–	Emission index of sulfur dioxide	$[\mathrm{g/\ kg}]$
FB	–	Fuel burn	$[\mathrm{kg}]$
FF	–	Fuel flow	$[\mathrm{kg/\ s}]$
FF_c	–	Corrected fuel flow	$[\mathrm{kg/\ s}]$
FF_{cruise}	–	Cruise fuel flow	$[\mathrm{kg/\ s}]$
FF_{min}	–	Minimum fuel flow	$[\mathrm{kg/\ s}]$
g	–	Gravity acceleration	$[\mathrm{m/\ s^2}]$

GC	–	Green cost	$[€]$
h	–	Altitude	$[\mathrm{m}]$
H	–	Hamiltonian	N/A
H_p	–	Geo-potential pressure altitude	$[\mathrm{ft}]$
$H_{p,\mathrm{des}}$	–	Transition altitude	$[\mathrm{ft}]$
HC	–	Hydrocarbons	$[\mathrm{g}]$
$\mathrm{H_2O}$	–	Water vapor	$[\mathrm{g}]$
j	–	Iteration index	N/A
J	–	Performance index	N/A
L	–	Lagrange term of performance index	N/A
Lft	–	Lift	$[\mathrm{N}]$
m	–	Aircraft mass	$[\mathrm{kg}]$
M	–	Mach number	N/A
N	–	Number of time stages	N/A
NO_x	–	Oxides of nitrogen	$[\mathrm{g}]$
P	–	Target	N/A
P	–	Air pressure	$[\mathrm{Pa}]$
P_0	–	Standard atmospheric pressure	$[\mathrm{Pa}]$
P_v	–	Saturation vapor pressure	$[\mathrm{Pa}]$
r	–	Region size	N/A
R	–	Real gas constant for air	$\left[\mathrm{m}^2/\left(\mathrm{K\,s}^2\right)\right]$
R^2	–	Coefficient of determination	N/A
RAE_G	–	Global relative approximation error	N/A
RAE_T	–	Training relative approximation error	N/A
RAE_{test}	–	Test relative approximation error	N/A
REI_{CO}	–	Reference emission index of carbon monoxide	$[\mathrm{g/\,kg}]$
REI_{HC}	–	Reference emission index of hydrocarbon	$[\mathrm{g/\,kg}]$
$REI_{\mathrm{NO_x}}$	–	Reference emission index of oxides of nitrogen	$[\mathrm{g/\,kg}]$
S	–	Number of grid points	N/A

S	–	Wing surface area	$\left[\mathrm{m}^2\right]$
SH_c	–	Humidity correction factor	$\left[\mathrm{kg/kg}\right]$
SO_2	–	Sulfur dioxide	$\left[\mathrm{g}\right]$
$t \in \mathbb{R}$	–	Time	$\left[\mathrm{s}\right]$
T	–	Engine thrust	$\left[\mathrm{N}\right]$
T	–	Air temperature	$\left[\mathrm{K}\right]$
T_0	–	Standard atmospheric temperature	$\left[\mathrm{K}\right]$
T_{cruise}	–	Cruise thrust	$\left[\mathrm{N}\right]$
$T_{\text{des,high}}$	–	High-altitude descent thrust	$\left[\mathrm{N}\right]$
$T_{\text{des,low}}$	–	Low-altitude descent thrust	$\left[\mathrm{N}\right]$
T_{maxclimb}	–	Maximum climb thrust	$\left[\mathrm{N}\right]$
$\left(T_{\text{maxclimb}}\right)\mathrm{ISA}$	–	Maximum climb thrust at standard atmosphere	$\left[\mathrm{N}\right]$
$T_{\text{max cruise}}$	–	Maximum cruise thrust	$\left[\mathrm{N}\right]$
$u_1(t)$	–	Acceleration	$\left[\mathrm{m/s}^2\right]$
$u_2(t)$	–	Flight path angle rate	$\left[\mathrm{rad/s}\right]$
$u_3(t)$	–	Heading rate	$\left[\mathrm{rad/s}\right]$
$U(t) \in \mathbb{R}^m$	–	Control vector	N/A
$V(t)$	–	Velocity	$\left[\mathrm{m/s}\right]$
V_{TAS}	–	True airspeed	$\left[\mathrm{m/s}\right]$
V_{CAS}	–	Calibrated airspeed	$\left[\mathrm{m/s}\right]$
WP	–	Waypoint	N/A
(x, y, z)	–	Geocentric coordinates	$\left[\mathrm{m}\right]$
$X(t) \in \mathbb{R}^n$	–	State vector	N/A
$Y(t)$	–	Combined state and co-state vector	N/A

Observation:

Although some symbols represent more than one quantity, these were intentionally left as such for the sake of convenience, since the symbol in question is commonly used in the literature to

describe the respective quantity. Meanwhile, all the symbols are duly listed throughout the document, along with the respective expressions.

Chapter 1

1. Introduction

1.1 Motivation

Improving aircraft operational efficiency has become a dominant topic in today's air transportation system, as airlines around the world have seen the rise of jet fuel prices during the past decade. In 2019, the global aviation industry has spent a total of $188 billion in jet fuel, which is approximately 23.7% of their total operating cost, this is an increase of around 4.7% over the 2018 fuel bill[1].

Moreover, it is estimated that the global aviation industry accounts for about 2% of total man-made global CO_2 emissions (in 2019, aviation has produced a total of 915 million tonnes CO_2, out of 43 billion tonnes CO_2 produced by the human) including international and domestic aviation and about 12% of the CO_2 from all transportation sources. In 2019, a total of 4.5 billion passengers were carried by the world's airlines, in comparison to 4.3 billion passengers in 2018. Although in 2020 the air traffic decreased due to global pandemic, yet for the next 20 years, the air traffic growth forecast to increase by an average of 4.3% per year. Therefore, the aviation sector will play a major role to increase global warming[2].

This increase in fuel cost and environmental concerns have pushed airlines to reduce fuel consumption and aircraft emissions, and to find margins for performance improvements. In 2009, the aviation industry has set a goal to reduce net aviation CO_2 emissions by 50% by 2050, relative to the 2005 levels. In order to achieve this goal, the industry is pursuing a four-pillar strategy[3]:

- New technology in aircraft design (e.g., efficient engines, lighter materials, new aerodynamics design, etc.), including the deployment of sustainable alternative fuels (e.g., biofuels).
- More efficient aircraft operations (e.g., trajectory-based operation).
- Infrastructure improvements, including modernized Air Traffic Management (ATM) system.

[1] Source: International Air Transport Association (IATA). https://www.iata.org/en/iata-repository/pressroom/fact-sheets/fact-sheet---fuel/.
[2] Source: Air Transport Action Group (ATAG). https://www.atag.org/facts-figures.html
[3] Source: International Air Transport Association (IATA). https://www.iata.org/en/iata-repository/pressroom/fact-sheets/fact-sheet---climate-change/

- A single Global Market-Based Measure (GMBM) to fill the remaining emissions gap.

Among those four-pillar strategies "more efficient aircraft operations" is appealing due to its rapid implementation timeline. On the other hand, efforts to modernize the aircraft fleet are limited by an extremely slow and expensive process of new aircraft adoption, which can take decades. Hence, efficient aircraft operations can be used to reduce the fuel consumption and environmental impact in current aircraft, which will likely share the sky with most modern aircraft in near future.

Aircraft trajectory optimization is an essential component to improve flight operation efficiency and to enhance the air traffic capacity, this technique not only helps to reduce the operational costs (e.g., fuel and time-related costs) but also helps to reduce the environmental impact (e.g., emissions, and noise, etc.) caused by the airliners.

To improve the ATM system under International Civil Aviation Organization (ICAO), In 2004, European Union (EU) launched the Single European Sky ATM Research (SESAR) program[4] and the USA launched the Next Generation Air Transportation System (NextGen) program[5]. One of the main visions of these programs is the initiation of an automatic system based on the Trajectory-Based Operation (TBO) concept. The TBO concept introduces time-based management, where the aircraft's current and planned position is known and shared, so the aircraft can be handled by their trajectory. The four-dimensional (4D) trajectory is needed to realize the TBO, where the 4D trajectory of an aircraft consists of the three spatial dimensions (x, y, h) plus time t as the fourth dimension. Fig. (1.1) illustrates a 4D trajectory of an aircraft.

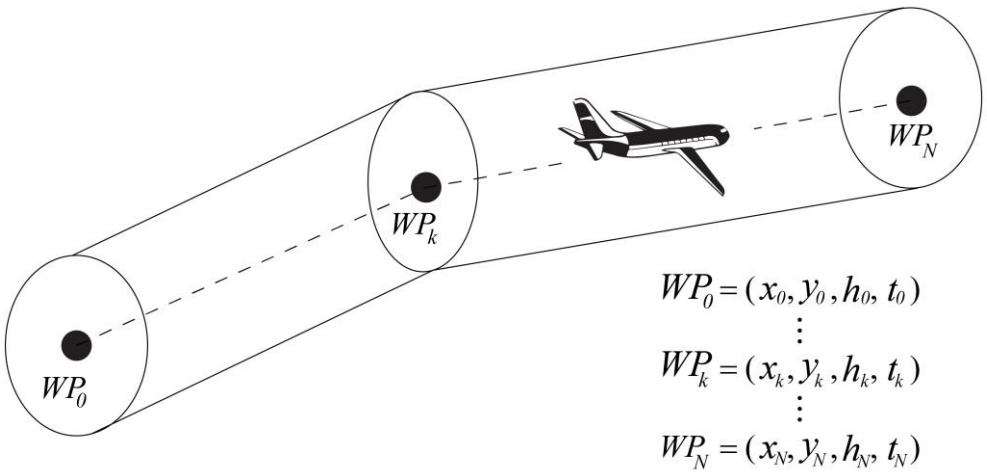


Figure 1.1: Representation of a 4D trajectory.

[4] Source: Single European Sky ATM Research (SESAR). https://www.sesarju.eu/

[5] Source: Next Generation Air Transportation System (NextGen). https://www.faa.gov/nextgen/

The existing flight planning techniques are mostly suboptimal and most of the flights do not fly at optimal speed or altitude, which increases the fuel consumption [1]. This situation can be encouraged by several foreseen or unforeseen factors (e.g., meteorological conditions, air traffic regulation limits, the horizontal and vertical spacing between aircraft, etc.). The 4D TBO is an appealing concept to solve this issue. As the aircraft are handled by their trajectory in time-based management, these foreseen or unforeseen factors can be dealt with effortlessly by using this concept.

Generally, the 4D optimal trajectory will be determined on the ground and inputted to the aircraft's Flight Management Computer (FMC) as a flight plan. Then the FMC can guide the aircraft along with the 4D optimal trajectory. In the case of a sudden change of flight plan, the FMC will also be able to solve the online Trajectory Optimization Problem (TOP) using efficient methods and find a new 4D optimal trajectory.

The underlying motivation of this thesis is to conduct research towards the aircraft 4D trajectory optimization dealing with fuel-saving and environmental impact reduction applied to the real implementation of TBO.

1.2 Background

In this section, a literature review of aircraft trajectory optimization that deals with fuel-saving and environmental impact reduction is addressed. Firstly, the prior works that have been done on aircraft trajectory optimization by applying different numerical approaches (e.g., Indirect method, direct method, dynamic programming, etc.) to reduce fuel consumption and environmental impact are presented. Following, the prior works of fuel-saving and environmental impact reduction on different flight scenarios are presented.

The aircraft Trajectory Optimization Problem (TOP) can be formulated as an Optimal Control Problem (OCP), where it is desired to find the state trajectory and the corresponding control inputs that optimize a specified performance measure while satisfying any constraints on the Equations of Motion (EOMs). For most practical applications, the TOPs are solved by numerical approaches. Typically, the numerical approaches to solve TOP can be classified into indirect methods and direct methods [2], [3], however, another class of numerical approach called Dynamic Programming (DP) is also considered extensively to solve the aircraft TOP.

In the indirect methods, the TOP is converted into Boundary Value Problem (BVP) by analytically formulating the first-order necessary condition for optimality which is derived from Pontryagin's Maximum Principle (PMP). The main advantages of indirect methods are that they lead to high-accuracy solutions and guarantee that the solution satisfies the optimality condition. However, it requires a good initial approximation of the co-state, which is difficult to guess. Besides, the BVPs that arise for many practical TOPs in indirect methods are quite difficult to solve, because of the complex dynamics and constraints structure of the problem.

Several works have been done to solve the aircraft TOPs by indirect methods. Burrows showed the application of PMP to generate fuel optimal trajectory with fixed arrival times [4]. Neuman and Kreindler computed the three-dimensional minimum fuel flight path using the PMP [5]. Chakravarty applied PMP to solve the four-dimensional (4D) fuel-optimal trajectory in the presence of winds for the cruise and descent phase [6]. Schultz applied a method based on Euler-Lagrange optimization theory to solve the minimum-time three-dimensional aircraft trajectory optimization problem by optimizing a modified Hamiltonian [7]. Barron and Chick III presented an improved indirect method to solve the two-point boundary value trajectory optimization problem, where the ascent and descent maneuvers of an Unmanned Air Vehicle (UAV) were optimized [8].

On the other hand, the direct methods discretize the infinite-dimensional original TOP into finite-dimensional Nonlinear Programming (NLP) problem, which is then solved numerically by the well-established optimization techniques. At present, the direct methods are widely used for solving TOPs, since not only these methods do not require an analytic expression for the necessary conditions of optimality, which can be a daunting task for complicated nonlinear dynamics, but also, they tend to have better convergence properties over indirect methods. Another great advantage of direct methods is that they do not need an initial guess of co-states, as it is difficult to predict the time histories.

Direct methods have been used extensively to solve aircraft TOPs. Jansch and Paus solved aircraft trajectory optimization using movable nodes direct collocation [9]. Raivio *et al.* compared the direct collocation and differential inclusion methods for calculating fighter aircraft optimal trajectories [10]. Ahmed *et al.* presented a spline parameterization technique to generate minimum length optimal trajectory along with 4D waypoints [11]. Bousson and Gameiro presented a quintic spline approach for UAV 4D trajectory generation [12]. Soler *et al.* used Hermite–Simpson collocation method to solve the TOP dealing with optimal takeoff weight and minimum fuel consumption [13]. Tian *et al.* optimized commercial 4D flight for both conventional and environmental cost using the A* algorithm and trapezoidal collocation method [14].

Recently, some researchers have been using Pseudospectral (PS) collocation which uses high-order orthogonal polynomials to solve aircraft TOP. Houacine and Khardi 2010 applied Gauss Pseudospectral Method (GPM) to generate an optimal trajectory that reduces noise and fuel consumption [15]. Bittner *et al.* proposed a Multi-Model GPM for both the Three Degree of Freedom (3DOF) and Six Degree of Freedom (6DOF) model of aerobatic aircraft trajectory [16]. Bousson and Machado applied Chebyshev Pseudospectral Method (CPM) to solve 4D flight trajectory optimization [17].

Aside from direct and indirect methods, Dynamic Programming (DP) is another well-established numerical method to solve TOP [18]. The numerical framework of DP is very suitable to handle discrete-time dynamic systems with nonlinear characteristics. Moreover, the 4D waypoints representation of the flight trajectory is similar to the discretization of the states grid system,

consequently, DP is a natural numerical method to deal with the 4D flight trajectory optimization. More great advantage of using DP is that it not only guarantees absolute (global) optimum but also can easily handle equality and inequality constraints of the system.

Miyamoto *et al* showed that the DP can be successfully used to solve fuel-efficient optimal aircraft flight trajectories [19]. Waller *et al.* applied the DP to solve a terrain-following aircraft optimization problem [20]. Grabbe *et al.* investigated the benefit of user-preferred trajectories by using a DP algorithm capable of determining the wind-optimal trajectory minimizing the flight time [21]. Hagelauer and Mora Camino presented a soft dynamic programming approach by using a neural network to solve online aircraft 4D trajectory optimization [22]. Miyazawa *et al.* proposed a Moving Search space Dynamic Programming (MS-DP) to reduce the computation time of traditional DP and applied it to the generation of conflict-free and minimum fuel 4D optimal trajectory [23]. Harada *et al.* proposed a method by using the piecewise linear approximation to overcome the limitation of the menace of the expanding grid problem of DP [24]. Khardi showed that the continuous analog of DP the Hamilton-Jacobi-Bellman (HJB) equation is well-suited for aircraft trajectory optimization problems and could minimize aircraft noise, fuel consumption, and air pollution around airports [25]. Parzani and Puechmorel applied the HJB approach to generate conflict-free minimum-time aircraft trajectory [26].

In the 1990s, a class of DP called Iterative Dynamic Programming (IDP) was proposed by luus, which solve the menace of expanding grid problem and showed better performance than the traditional DP [27]. Later on, the IDP was extended to Single Grid-point Dynamic Programming (SGDP) which can be used to solve online TOP for collision avoidance with accuracy [28].

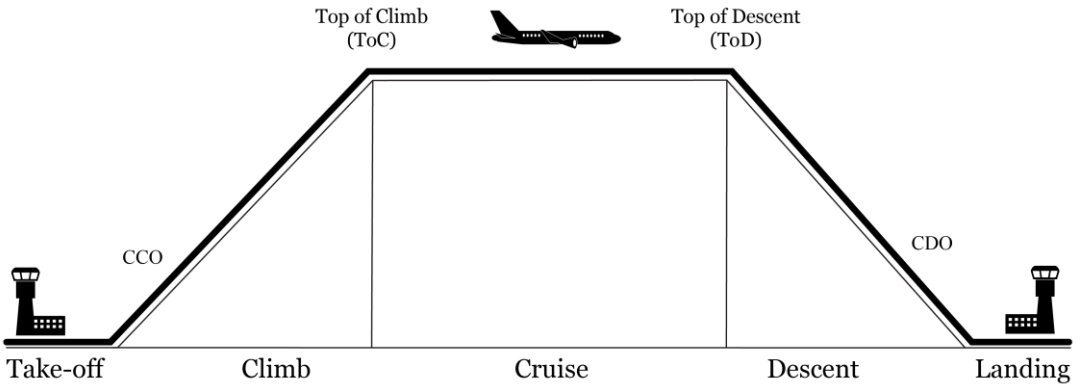


Figure 1.2: Three main flight phases.

In order to optimize an aircraft trajectory, one must consider that it is composed of different flight phases (e.g., take-off, climb, cruise, descent, approach, etc.). Generally, the three main flight phases (i.e., climb, cruise, and descent phases) are considered in most of the prior research on aircraft trajectory optimization that deals with fuel consumption and environmental impact reduction, some researchers optimized the aircraft trajectory considering just one phase, others

considered the global trajectory consist of all the phases. Fig. (1.2) illustrates the typical flight phases of commercial flights.

Recently, an operational technique for the climb phase called Continuous Climb Operation (CCO) has been introduced in the Air Traffic Management (ATM) research. Unlike the conventional departure, the CCO procedure allows departing aircraft to climb continuously to attain the Top of Climb (ToC) at optimum climb engine thrust and climb speed without intermediates level-off, which leads to a significant reduction of fuel burn and environmental impact in the climb phase. This CCO procedure is also known as the green departure. Several recent studies [34], [35], [36] showed the benefits of using the CCO procedure for fuel-efficiency. Fig. (1.3) illustrates the CCO procedure, where the blue line shows the CCO, and the dotted line shows the conventional step climb procedure.

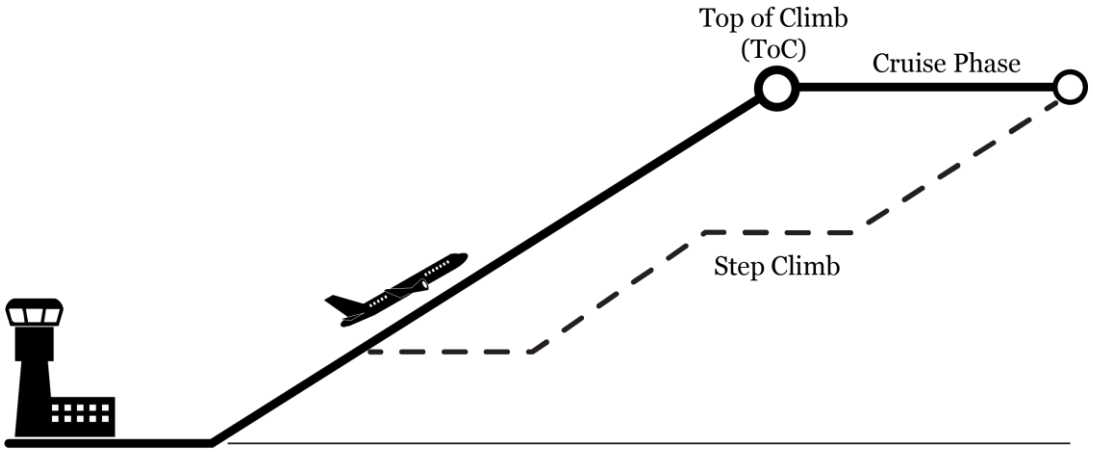


Figure 1.3: Illustration of continuous climb operation.

The cruise phase is the time period following the climb phase when the aircraft is in level flight. In general, most fuel is burnt during the cruise phases of flight except for short-haul flights; therefore, the trajectory optimization of the cruise phase of flight has got special attention in the literature. Speyer and Schultz are the one of the first to work on fuel optimality in the cruise phase, assuming constant aircraft mass [37], [38]. Shepard and Bilimoria investigated the optimization of aircraft cruise performance by considering three cases, constant altitude with variable velocity, constant velocity with variable altitude, and variable altitude with variable velocity [39]. Liden presented methods for computing optimum cruise profile with step climb points in the presence of winds [40].

Recently, [41], [42] have done some studies to design fuel-optimal trajectory during the cruise phase at a constant altitude, Franco and Rivas analyzed the same problem including wind effects [43]. Jensen *et al.* analyzed the fuel efficiency benefits through cruise speed optimization in high altitude commercial flight [1]. Delgado and Prats also showed how the cruise speed variations

affect the fuel consumption in the cruise phase of flight [44]. Pawlak *et al.* applied DA to minimize NO_x, CO, HC, and CO_2 emissions of a passenger aircraft in a cruise phase [45]. Tian *et al.* established an aircraft green direct operating cost (GDOC) model to reduce aircraft emissions in the cruise phase by using DA [46].

The descent phase is the time period when the aircraft has a negative rate of climb for an extended period of time, descent phase trajectory optimization of commercial aircraft is important to minimize the economic and environmental impact. In literature, the noise abatement in the descent phase has been considered heavily. Fogarty and Howe in the late 1960s solved a step-by-step descent trajectory optimization for minimum cost using the direct gradient method [47]. Park and Clarke studied the descent phase trajectory optimization in presence of wind, minimizing environmental impacts, where both the fuel and emissions costs were considered [48].

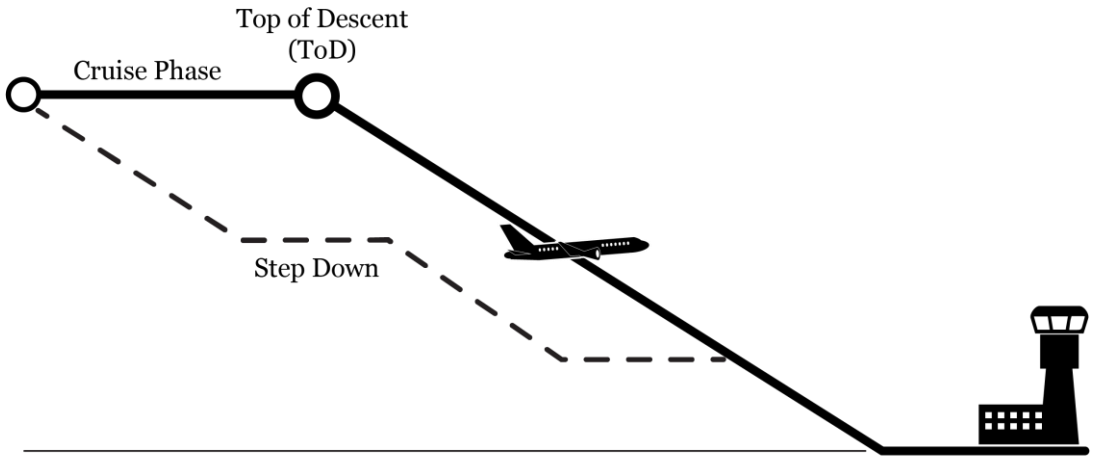


Figure 1.4: Illustration of continuous descent operation.

An operation concept known as Continuous Descent Operation (CDO) has led to significant fuel saving and environmental benefits during the descent phase, unlike the conventional approach, the CDO procedure allows arriving aircraft to stay higher for longer and descend continuously. In this procedure, the aircraft employ minimum engine thrust, ideally from Top of Descent (ToD) and in a low drag configuration prior to the final approach fix without intermediates level-off. As CDO requires significantly less engine thrust than the conventional approach and the aircraft stays higher for longer, it results in less fuel burn, less emission, and less noise. Fig. (1.4) illustrates the CDO procedure compared with the conventional step-down procedure.

Wubben and Busink analyzed the environmental benefits of the CDO approach based on FMS data of actual flights and compared the results with the conventional approach procedures [49].

Park and Clarke investigated the multiphase trajectory generation problem based on the CDO approach to minimize flight time and fuel consumption [50]. Cao *et al.* evaluated the CDO approach as a standard terminal airspace operation to save fuel [51]. Tian *et al.* minimized the aircraft emissions and noise level in the terminal area by incorporating the CDO approach [52].

Besides considering only individual phases of flight for trajectory optimization lots of researchers considered the global trajectory optimization, where typically the global trajectory consists of the climb, cruise, and descent phases. In the late 1970s, several works have been published that considered global trajectory optimization problems including the climb, cruise, and descent phases to compute minimum-fuel, minimum-time, and minimum Direct Operation Cost (DOC) for commercial aircraft trajectories [53], [54].

Recently, Rivas *et al.* presented a trajectory computational tool to compute global trajectories of commercial transport aircraft [55]. The tool employed a kinetic modeling approach and the aircraft performance model, and the formulation considers wind effects and temperature corrections for a non-standard atmosphere. Patrón *et al.* presented a 3D trajectory optimization algorithm to minimize the global cost for the climb, cruise, and descent phases of the flight in the presence of wind [56]. Celis *et al.* developed an optimization algorithm based on the Genetic Algorithm (GA) capable of minimizing flight time, fuel consumption, and NO_x emissions in one or more flight phases [57]. Ahmed and Bousson applied DA to generate time-optimal and fuel-optimal global trajectories from 4D predefined waypoint networks [58], [59]. Tian *et al.* optimized the global 4D trajectory of commercial flight by taking into account both the conventional (fuel cost and time cost) and the environmental cost (greenhouse gas cost and harmful gas cost) [14].

1.3 Research Objectives

The main goal of this thesis is to develop a trajectory optimization algorithm to solve aircraft 4D trajectory optimization problems for fuel-saving and environmental impact reduction. To reach this goal the following objectives must be achieved:

- Brief description of the literature review and formulation of aircraft trajectory optimization problem. The formulation includes the aircraft dynamic model, performance model, and emissions model.
- Brief description of the literature review on numerical methods to solve the trajectory optimization problem. The review includes the three basic classes of methods (e.g., indirect method, direct method, and dynamic programming).
- Proposing efficient methods to solve aircraft trajectory optimization problems for different cases.
- Validation of the proposed method by solving numerical examples dealing with fuel-saving and environmental impact reduction (aircraft emissions). The example includes

the climb phase, cruise phase, descent phase, and global trajectory which consists of all three phases that were discussed above.

- Predict commercial aircraft fuel flow rate from actual flight data by using an artificial neural network and to apply the fuel flow rate model to generate an optimal trajectory that minimizes the green cost (fuel and emissions cost).

1.4 Outline of the Thesis

This thesis consists of a total of eight chapters and two annexes organized as follow:

In chapter 2, the general formulation of the aircraft Trajectory Optimization Problem (TOP) is presented. The first three sections present the three key steps of the aircraft TOP (e.g., mathematical model, Performance Index (PI), and boundary and path constraints). The mathematical model includes aircraft performance (atmosphere, aerodynamic, engine thrust. and fuel consumption and emissions model). Finally, at the end of the chapter, a general formulation of aircraft TOP is presented.

In chapter 3, the three basic classes of numerical methods (e.g., indirect method, direct method, and dynamic programming) to solve TOP are presented. Various techniques of each method along with their characteristics, advantages, and disadvantages are also presented.

In chapter 4, the generation of fuel and time-optimal trajectories from predefined 4D waypoint networks by applying a single source shortest path algorithm is presented. The algorithm was validated by three case studies with different lengths of flights (short, medium, and long-haul flights).

In chapter 5, the generation of minimum length optimal trajectories along predefined 4D waypoints is presented. The original TOP is transformed into a Nonlinear Programming (NLP) problem by the cubic spline interpolation, then an NLP solver is used to solve the resulting NLP and to determine the 4D optimal trajectory. Two numerical examples (take-off phase and cruise phase) are presented to validate the approach.

In chapter 6, a Modified Dynamic Programming (MDP) approach is proposed to solve the TOP. The MDP approach reduces the computational burden and overcomes the drawbacks of the traditional DP while retaining its appealing features. The proposed approach was validated by a total of eight numerical examples containing the climb phase, cruise phase, descent phase, and global trajectory for fuel optimal and emissions optimal trajectories.

In chapter 7, the Radial Basis Function (RBF) neural network was applied to predict the aircraft fuel flow rate for the climb, cruise, and descent phases of flight. Afterward, the predicted fuel flow rate model was used to generate an aircraft optimal 4D green trajectory that optimizes the green cost of a commercial flight trajectory.

In chapter 8, a summary of the research work, conclusions, contributions, and future works are discussed.

In annex A, the performance operational data of two aircraft are presented.

In annex B, the relevant journal publications and conference communications of the thesis are presented.

Chapter 2

2. Formulation of Aircraft Trajectory Optimization Problem

A Trajectory Optimization Problem (TOP) is a class of Optimal Control Problems (OCP) where the main objective is to optimize a measure of performance (e.g., minimum time, minimum fuel consumption, etc.) over a trajectory of a vehicle (e.g., aircraft, spacecraft, satellite, etc.) while satisfying a set of constraints.

This chapter presents the general formulation of the aircraft TOP. The formulation of TOP can be described by three general steps. The steps include mathematical modeling of the system dynamics, defining appropriate performance index (i.e., objective function), and the boundary and path constraints on the states and controls of the system.

In the first section, the aircraft Equations of Motion (EOMs) based on the Three Degree of Freedom (3-DOF) are presented to model the aircraft dynamics, then the aircraft performance models (e.g., atmosphere model, aerodynamic model, engine thrust model, and fuel consumption model, etc.) are described. Subsequently, the aircraft emissions models are also presented. The next section presents a couple of performance indexes that can be used in aircraft TOP. The third section shows some typical boundary and path constraints that are imposed on the aircraft in TOP. Finally, at the end of the chapter, a general formulation of TOP is presented.

2.1 Mathematical Model

The first step to formulate the aircraft Trajectory Optimization Problem (TOP) refers to the mathematical modeling of the problem. In this chapter, mathematical modeling is described in three parts. The first part is the aircraft dynamics model where a set of nonlinear Equations of Motion (EOMs) are derived to relate the different forces that describe the motion of the aircraft. The second part is the aircraft performance model where the engine and aerodynamics characteristics of the aircraft are described. In the third part, the aircraft emissions models are described.

2.1.1 Aircraft Dynamics Model

Generally, the aircraft system dynamics is modeled by a set of nonlinear Equations of Motion (EOMs), which are referred to as the state or system equations representing a path or time history of the position and velocity of the aircraft.

Typically, the aircraft EOMs are composed of translational equations (e.g., position, velocity, acceleration, forces, etc.) and rotational equations (e.g., moments, angular velocity, angular acceleration, etc.) which are also known as the Six Degree of Freedom (6-DOF) EOMs. However, for the aircraft Trajectory Optimization Problem (TOP), it is assumed that the aircraft rotational rates are small, and its control surface deflection does not affect aerodynamic forces. So, a common assumption in aircraft trajectory optimization is that the EOMs are only composed of the translational equations, which are also known as the Three Degree of Freedom (3-DOF) EOMs [60], [61], [62].

In order to derive the EOMs for a non-steady flight of an aircraft, some assumptions and reference coordinate systems need to be defined. The following assumptions were made to derive the EOMs:

- The aircraft is assumed to be a variable mass rigid body, with fixed engines.
- The aircraft has a plane of symmetry (i.e., The forces act at the center of gravity and the thrust, and the aerodynamic forces lie on the plane of symmetry)
- For aircraft motion, the earth is an approximate inertial reference frame, flat and nonrotating, the model is called the flat earth model.
- The acceleration of gravity in the atmospheric flight is constant and perpendicular to the surface of the earth.
- The atmosphere is at rest relative to the earth, and the atmospheric properties are functions of altitude only.

After the above assumptions were made, to derive the EOMs total of four reference coordinate systems need to be defined as follow:

- *Earth reference frame:* An earth reference frame (E, x, y, h) is fixed to the surface of the earth at mean sea level. Typically, the x axis lies in the horizontal plane and points towards the east, y axis points towards the north, and the h axis points opposite to the center of the earth. This reference system is an approximate inertial reference frame.
- *Local reference frame:* The local reference frame (O, x_h, y_h, h_h) moves with the aircraft center of gravity O. However, this frame always remains parallel to the Earth reference frame.
- *Wind reference frame:* The wind reference frame (O, x_w, y_w, h_w) also moves with the aircraft center of gravity O. Where the x_w axis of this frame coincides with the aerodynamic velocity vector of the aircraft.

- *Body reference frame:* The body reference frame (O, x_b, y_b, h_b) is fixed to the aircraft's center of gravity O. Where the x_b axis of this frame coincides with the fuselage centerline, the h_b axis is perpendicular to the x_b axis, and the y_b axis is perpendicular to the x_b, h_b plane.

These four reference coordinate systems for flight are shown in Fig. (2.1), where the aircraft is located at an altitude h above the mean sea level and (x, y) coordinate points. The velocity V of the aircraft is relative to the air [63].

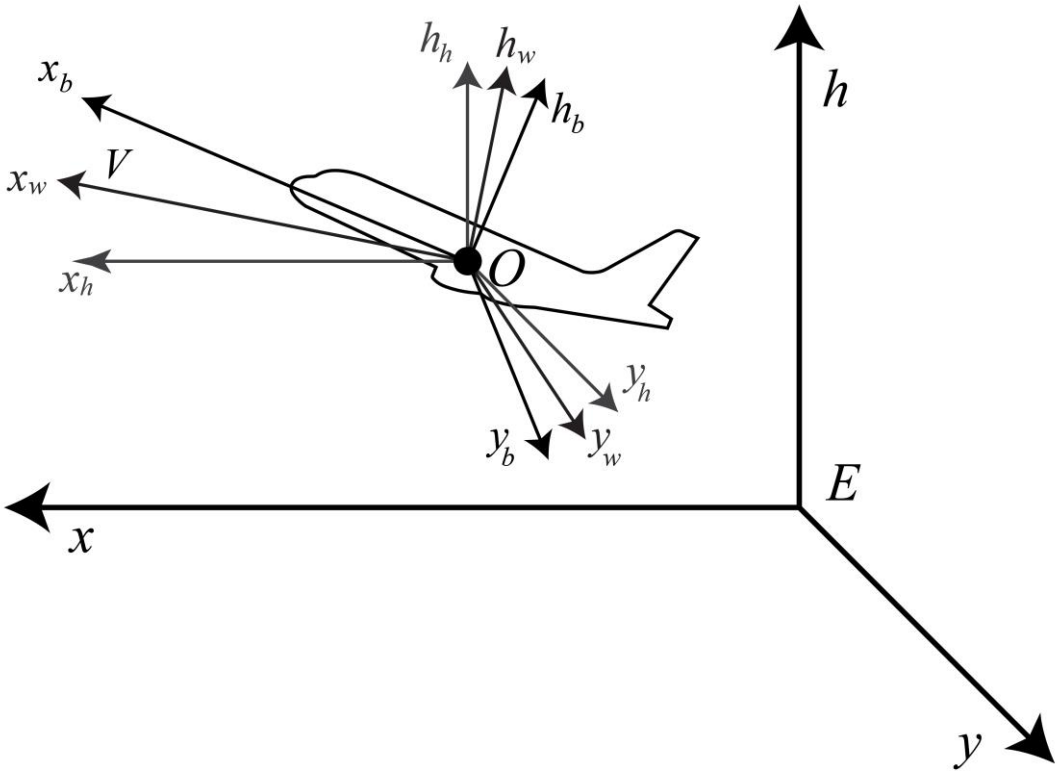


Figure 2.1: Reference coordinate systems for flight.

By using the above assumptions and reference coordinates, now the 3-DOF EOMs can be defined by the following differential equations:

$$\dot{x} = V \cos\gamma \cos\psi \tag{2.1}$$

$$\dot{y} = V \cos\gamma \sin\psi \tag{2.2}$$

$$\dot{h} = V \sin\gamma \tag{2.3}$$

$$\dot{V} = g\left(\frac{T\cos\alpha - D}{mg} - \sin\gamma\right) \tag{2.4}$$

$$\dot{\gamma} = \frac{1}{mV}\left(\left(T\sin\alpha + Lft\right)\cos\mu - mg\cos\gamma\right) \tag{2.5}$$

$$\dot{\psi} = \frac{\left(T\sin\alpha + Lft\right)\sin\mu}{mV\cos\gamma} \tag{2.6}$$

$$\dot{m} = -FF \tag{2.7}$$

(x, y, h) = Three-dimensional position of the aircraft;

V = Velocity;

γ = Flight path angle;

ψ = Heading angle;

m = Aircraft mass;

α = Angle of attack;

T = Thrust force;

μ = Bank angle;

D = Aerodynamic drag;

Lft = Aerodynamic lift;

g = Gravity acceleration;

FF = Fuel flow;

The set of EOMs Eqs. (2.1) – (2.7) describe the 3DOF system (i.e., the system allows three variables that can be used as control variables). Generally, the state vector of the flight is composed of the variables whose derivatives appear in the EOMs $X(t) = \left[x(t), y(t), h(t), V(t), \gamma(t), \psi(t), m(t)\right]$, and the control vector is composed of the following variables $U(t) = \left[\mathrm{T}(t), \alpha(t), \mu(t)\right]$. However, the control variables to choose may vary. The forward movement of the aircraft can be implemented by using one of these variables: thrust, engine power settings, or acceleration, in any case, the concept is the same, which can be controlled by the engine throttle. Similarly, the vertical movement of the aircraft can be implemented using either, the angle of attack, lift force, the lift coefficient, or the flight path angle rate, which is generally controlled by the elevator trims. And finally, the sideways movement of the aircraft can be implemented by one of the following variables: the bank angle or the heading rate, which is controlled by combining the rudder and ailerons trim.

Fig. (2.2) shows the four main forces acting on the aircraft. These are the engine thrust T, which acts along the body axis x_b. The aerodynamic lift Lft, that acts perpendicular to the wind axis x_w. The induced drag D, which acts opposite to the forward velocity axis. And the aircraft's

weight W acts vertically downwards to the center of the earth. Fig. (2.2) also shows the angle of attack α, flight path angle γ, and heading angle ψ of the aircraft. Where the angle of attack α is the angle between the wind axis x_w and body axis x_b, the flight path angle γ is the angle between the wind axis x_w and local horizon axis x_h, and the heading angle ψ is the angle between the ground velocity V_g and local horizon axis x_h.

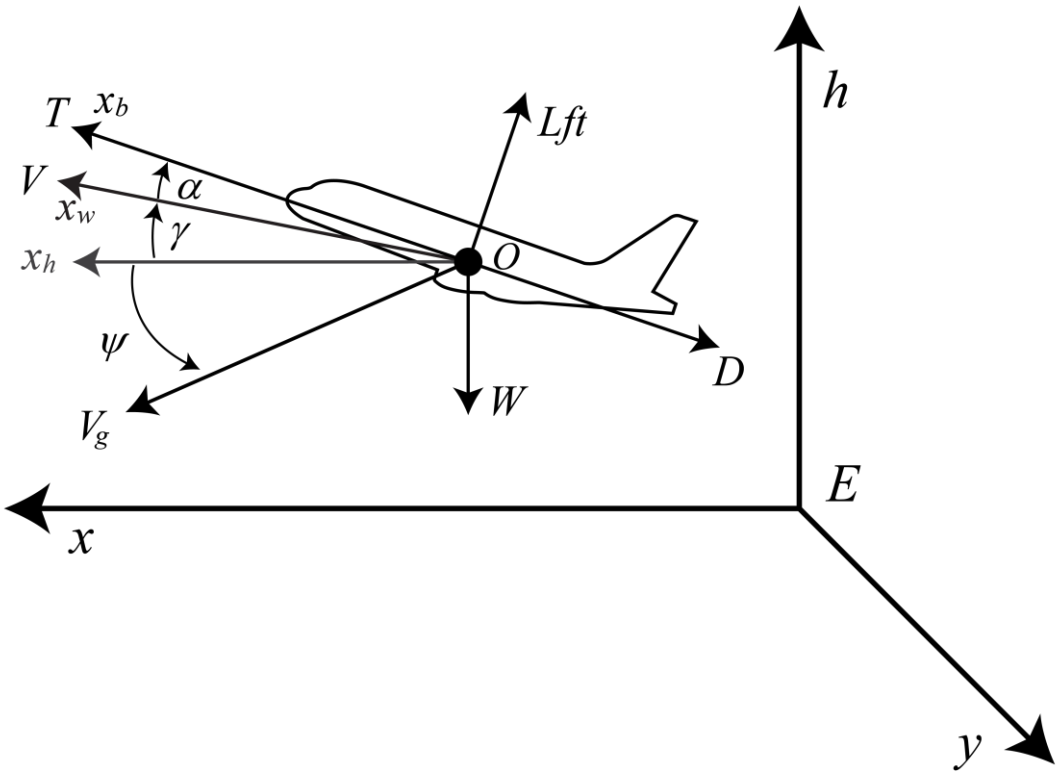


Figure 2.2: Forces acting on an aircraft in flight.

2.1.2 Aircraft Performance Model

The set of differential equations Eqs. (2.1) – (2.7) represent the 3DOF equation of motion for any aircraft. However, for the complete description of the aircraft motion, the aircraft performance (e.g., atmosphere, aerodynamic, engine thrust, and fuel consumption) models are required.

The Base of Aircraft Data (BADA) provides aircraft performance models that enable highly accurate aircraft trajectories modeling and simulations. BADA is maintained by EUROCONTROL through active cooperation with aircraft manufacturers [64]. In this thesis, a turbofan/turbojet engine subsonic aircraft is considered, and the atmosphere, aerodynamic, engine thrust, and fuel consumption model were extracted from BADA data.

2.1.2.1 Atmosphere Model

The atmosphere model provides expressions for air temperature, air pressure, air density, and speed of sound as a function of altitude. These atmospheric properties are required to convert the aircraft's true airspeed V_{TAS} to calibrated air speed V_{CAS} and Mach number M, as well as to calculate the other performance models.

A standard atmosphere condition is required to be defined in order to derive the full atmosphere model. The International Standard Atmosphere (ISA) condition occurs at the zero geo-potential pressure altitude $H_p = 0$, and they are denoted as:

Standard atmospheric temperature :	$T_0 = 288.15$ [K]
Standard atmospheric pressure :	$P_0 = 101325$ [Pa]
Standard atmospheric density :	$\rho_0 = 1.225$ [kg/m^3]
Speed of sound :	$a_0 = 340.294$ [m/s]

The atmosphere model has three layers. The first layer called the troposphere, which runs from the earth's surface to the geo-potential pressure altitude $H_p < 11000$ [m]. In this layer, the temperature decreases linearly, and the pressure decreases exponentially. When the geo-potential pressure altitude $H_p = 11000$ [m] is called the tropopause, which separates the troposphere and the lower stratosphere. The lower stratosphere is between the geopotential pressure altitude $11000 < H_p < 25000$ [m] and the upper stratosphere is when $H_p > 25000$ [m].

The air temperature is the highest near the surface of the earth and it decreases as the altitude increases. The air temperature T at any geo-potential pressure altitude H_P can be defined as follow:

$$T = T_0 + \Delta T + \beta_T H_P \tag{2.8}$$

Where ΔT is the temperature differential at Mean Sea Level (MSL) between a given non-standard atmosphere and ISA, $\beta_T = -0.0065$ is the ISA temperature gradient with an altitude below the tropopause.

The air pressure also decreases as the altitude increases. The air pressure P at any geo-potential pressure altitude H_P can be defined as follow:

$$P = P_0 \left(1 + \frac{\beta_T H_P}{T_0} \right)^{-\frac{g}{\beta_T R}} \tag{2.9}$$

Where $g = 9.80665$ is the gravity acceleration and $R = 287.05287$ is the real gas constant for air.

The air density depends on both the air temperature and the air pressure and decreases with increasing altitude. The air density ρ at any geo-potential pressure altitude H_P can be defined as follow:

$$\rho = \frac{P}{R \times T} \tag{2.10}$$

The speed of sound a at any geo-potential pressure altitude H_P can be defined as follow:

$$a = \sqrt{\kappa R T} \tag{2.11}$$

Where, $\kappa = 1.4$ is the adiabatic index of air.

Now by using the atmospheric model it is possible to calculate the calibrated air speed V_{CAS} as a function of true airspeed V_{TAS} as follow:

$$V_{CAS} = \left[\frac{2}{\left(\frac{\kappa - 1}{\kappa} \right)} \frac{P_0}{\rho_0} \left\{ \left(1 + \frac{P}{P_0} \left[\left(1 + \frac{\left(\frac{\kappa - 1}{\kappa} \right)}{2} \frac{\rho}{P} V_{TAS}^{\,2} \right)^{\frac{1}{\left(\frac{\kappa - 1}{\kappa} \right)}} - 1 \right] \right)^{\left(\frac{\kappa - 1}{\kappa} \right)} - 1 \right\} \right]^{\frac{1}{2}} \tag{2.12}$$

Also, the Mach Number M can be calculated as the function of true airspeed V_{TAS} and the air temperature as follow:

$$M = \frac{V_{TAS}}{\sqrt{\kappa R T}} \tag{2.13}$$

2.1.2.2 Aerodynamic Model

The aerodynamic lift Lft, and drag D are directly depend on the air density and commonly modeled by using lift and drag coefficient as follow:

$$Lft = \frac{1}{2} \rho V_{TAS}^2 C_L S \tag{2.14}$$

$$D = \frac{1}{2}\rho V_{TAS}^2 C_D \mathrm{S} \tag{2.15}$$

Where ρ is the air density, V_{TAS} is the true airspeed, S is the wing surface area, C_L is the lift coefficient, and C_D is the drag coefficient.

By assuming the flight path angle γ is zero and adding a correction for a bank angle μ, the lift coefficient C_L can be defined as follow:

$$C_L = \frac{2mg}{\rho V_{TAS}^2 \mathrm{S}\cos\mu} \tag{2.16}$$

Where, m is the mass of aircraft, g is the gravity acceleration, and $\frac{1}{2}\rho V_{TAS}^2$ is the dynamic pressure.

For different phases of flight, the drag coefficient C_D is specified as a function of lift coefficient C_L as follow:

$$C_D = C_{D0} + C_{D2}(C_L)^2 \tag{2.17}$$

Where C_{D0} is the parasite drag coefficient and C_{D2} is the induced drag coefficient. The values of these coefficients vary depending on the flap configuration in different phases of flight. However, in the landing phase, the parasite drag increases due to deployment of the landing gear, so the Eq. (2.17) becomes:

$$C_D = C_{D0} + C_{D0,\Delta LDG} + C_{D2}(C_L)^2 \tag{2.18}$$

Where the $C_{D0,\Delta LDG}$ is the increment of the parasite drag due to the landing gear. The BADA data provides the values of these coefficients for a number of specific aircraft in the Operations Performance File (OPF).

2.1.2.3 Engine Thrust Model

The engine thrust is calculated in Newtons for turbofan/turbojet engines. The BADA model provides coefficients that allow the calculation of different thrust levels based on different phases of flight as follow:

Maximum Climb and Take-off Thrust: The maximum climb thrust $(T_{\mathrm{maxclimb}})\mathrm{ISA}$ at standard atmosphere conditions is calculated as a function of geo-potential pressure altitude H_p as follow:

$$\left(T_{\text{maxclimb}}\right)\text{ISA}=C_{Tc,1}\left(1-\frac{H_p}{C_{Tc,2}}+C_{Tc,3}H_p{}^2\right) \tag{2.19}$$

Where, $C_{Tc,1}, C_{Tc,2}$, and $C_{Tc,3}$ are the climb thrust coefficients specified in the BADA OPF file. The Eq. (2.19) is the maximum climb thrust at standard atmosphere, which can be corrected for temperature deviation ΔT in the following way:

$$T_{\text{maxclimb}}=\left(T_{\text{maxclimb}}\right)\text{ISA}\left(1-C_{Tc,5}\left(\Delta\text{T}-C_{Tc,4}\right)\right) \tag{2.20}$$

Now by combining the Eqs. (2.19) and (2.20) the maximum climb thrust can be defined as follow:

$$T_{\text{maxclimb}}=C_{Tc,1}\left(1-\frac{H_p}{C_{Tc,2}}+C_{Tc,3}H_p{}^2\right)\times\left(1-C_{Tc,5}\left(\Delta\text{T}-C_{Tc,4}\right)\right) \tag{2.21}$$

Where, $C_{Tc,4}$ and $C_{Tc,5}$ are the thrust temperature coefficient specified in the BADA OPF file. The maximum climb thrust Eq. (2.21) is used for both take-off and climb phases.

Maximum Cruise Thrust: Generally, the thrust is set equal to drag in the cruise phase of flight as follow:

$$T_{\text{cruise}}=D_{\text{cruise}} \tag{2.22}$$

However, the maximum amount of thrust available in the cruise phase is limited, which is calculated as a ratio of the maximum climb thrust T_{maxclimb} as follows:

$$T_{\text{max cruise}}=C_{Tcr}\left(T_{\text{maxclimb}}\right) \tag{2.23}$$

Where C_{Tcr} is the maximum cruise thrust coefficient, its value is currently set to 0.95 for all aircraft.

Descent Thrust: Descent thrust is also calculated as a function of maximum climb thrust T_{maxclimb}, where different correction factor is used for high and low altitude descent. If the aircraft is flying above the transition altitude $H_{p,\text{des}}$, then the descent thrust is defined as follow:

$$T_{\text{des,high}}=C_{T\ \text{des,high}}\left(T_{\text{maxclimb}}\right) \tag{2.24}$$

If the aircraft is flying below the transition altitude $H_{p,\text{des}}$, then the descent thrust is defined as follow:

$$T_{\text{des,low}}=C_{T\ \text{des,low}}\left(T_{\text{maxclimb}}\right) \tag{2.25}$$

Where $H_{p,\text{des}}$ is the transition altitude, $C_{T\ \text{des,high}}$ is the high-altitude descent thrust coefficient, and $C_{T\ \text{des,low}}$ is the low-altitude descent thrust coefficient specified for a number of specific aircraft in the BADA OPF file.

2.1.2.4 Fuel Consumption Model

The fuel consumption of commercial flights depends on ambient temperature, true airspeed, and aircraft altitude. The BADA model provides the thrust-specific fuel consumption η that allows the calculation of the fuel consumption for turbofan/turbojet engines. The thrust specific fuel consumption η describes the fuel efficiency of an engine design with respect to thrust output, and is specified as a function of true airspeed V_{TAS} as follow:

$$\eta = C_{\text{f}1}\left(1 + \frac{V_{TAS}}{C_{\text{f}2}}\right) \tag{2.26}$$

Where, $C_{\text{f}1}$ and $C_{\text{f}2}$ are the thrust-specific fuel consumption coefficients specified for several specific aircraft in the BADA OPF file.

The nominal fuel flow FF can be calculated using the thrust T and thrust specific fuel consumption η as follow:

$$FF = \eta T \tag{2.27}$$

The thrust T in Eq. (2.27) varies with different flight phases. The nominal fuel flow FF Eq. (2.27) is used in all flight phases except during cruise and idle thrust descent, where the following expressions are to be used for the cruise phase of flight:

$$FF_{\text{cruise}} = \eta T C_{\text{f cr}} \tag{2.28}$$

Where, $C_{\text{f cr}}$ is the cruise fuel flow correction coefficient. For the moment the cruise fuel flow correction factor has been established for a number of aircraft types whenever the reference data for cruise fuel consumption is available and is specified in the BADA OPF file. This factor has been set to 1 (one) for all the other aircraft models.

During idle thrust descent the minimum fuel flow for turbofan/turbojet engines is specified as a function of the geopotential pressure altitude H_p as follow:

$$FF_{\text{min}} = C_{\text{f}3}\left(1 - \frac{H_p}{C_{\text{f}4}}\right) \tag{2.29}$$

Where, $C_{\text{f}3}$ and $C_{\text{f}4}$ are the descent fuel flow coefficients specified in the BADA OPF file.

2.1.3 Aircraft Emission Model

Principle greenhouse gas emissions resulting from aircraft in flight that impact the environment most are carbon dioxide (CO_2), water vapor (H_2O), sulfur dioxide (SO_2), oxides of nitrogen (NO_x), carbon monoxide (CO), and hydrocarbons (HC). Typically, these aircraft emissions are modeled by the Emission Index (EI), which has units of grams of emission per kilogram of fuel burned.

Although, the emissions of CO_2, H_2O and SO_2 are assumed to be proportional to the fuel consumption of the aircraft, the emissions of NO_x, CO and HC are influenced by a number of parameters, mostly by thrust setting of the engine, aircraft velocity, aircraft altitude, and ambient atmospheric condition.

2.1.3.1 CO_2, H_2O, and SO_2 Emissions Model

Carbon dioxide (CO_2) and water vapor (H_2O) are the primary greenhouse gas emissions produced by the combustion of jet fuel. The CO_2 emissions directly affect the climate, the H_2O emissions responsible for the formation of contrails and have a large contribution to the temperature rise due to aircraft emissions and the SO_2 emissions contribute to the acid rain. The CO_2, H_2O and SO_2 emission levels are determined by the fuel consumption and the respective quantity of carbon, hydrogen, and sulfur compounds contained in the fuel.

The carbon dioxide, water vapor, and sulfur dioxide emissions E_{CO_2}, E_{H_2O} and E_{SO_2} of commercial aircraft can be defined by the Emission Index (EI) of the respective greenhouse gases and fuel burn FB as follow:

$$E_{CO_2} = EI_{CO_2}\ FB \tag{2.30}$$

$$E_{H_2O} = EI_{H_2O}\ FB \tag{2.31}$$

$$E_{SO_2} = EI_{SO_2}\ FB \tag{2.32}$$

Where, EI_{CO_2}, EI_{H_2O} and EI_{SO_2} are respectively the emission Index of carbon dioxide, water vapor, and sulfur dioxide in grams and fuel burn FB in kilogram. The fuel burn can be defined as:

$$FB = FF \times t \tag{2.33}$$

Where FF is the fuel flow and t is the time.

Table 2.1: Emission indices for CO_2, H_2O and SO_2.

Emission	Emission Index [g/kg]
carbon dioxide (CO_2)	3155
water vapor (H_2O)	1237
sulfur dioxide (SO_2)	0.8

Source: (Sutkus *et al.* pp. 23, [65])

Table (2.1) shows the emission indices EI_{CO_2}, EI_{H_2O} and EI_{SO_2} which are based on the analysis of [66].

2.1.3.2 NO_x, CO, and HC Emissions Model

The emission indices of the oxides of nitrogen (NO_x), carbon monoxide (CO), and hydrocarbons (HC) vary with the engine combustor condition. Although the NO_x emissions do not directly contribute to climate change but NO_x interact with other atmospheric gases resulting in the formation of ozone and depletion of methane. The unburned CO and HC also affect the environment. Typically, the NO_x emission is the highest at the high thrust setting and the CO and HC emissions are the highest at the low thrust setting.

The emissions of the NO_x, CO and HC can be modeled by the Boeing Fuel Flow Method 2 (BFFM2). The BFFM2 uses the International Civil Aviation Organization (ICAO) emission data bank to determine the reference emission indices, which eventually allow the calculation of the emissions of these gases [67].

The first step to model the emissions of the NO_x, CO and HC is to correct the fuel flow by taking into account the ambient temperature, pressure, and Mach number.

$$FF_c = \left(\frac{FF}{\delta_{amb}} \right) \theta_{amb}^{3.8} \, e^{0.2M^2} \tag{2.34}$$

Where FF_c is the corrected fuel flow, M is the Mach number that can be calculated using Eq. (2.13). θ_{amb} is the temperature ratio and δ_{amb} is the pressure ratio, which can be defined as follow:

$$\theta_{amb} = \frac{T}{T_0} \tag{2.35}$$

$$\delta_{amb} = \frac{\mathrm{P}}{\mathrm{P}_0} \tag{2.36}$$

Where, $\mathrm{T}_0 = 288.15$ [K] and $\mathrm{P}_0 = 101325$ [Pa] are the standard atmospheric temperature and pressure. The T and P are the ambient temperature and pressure that can be calculated using Eqs. (2.8) and (2.9).

The BFFM2 uses the ICAO emission data bank to determine the reference emission index $REI_{\mathrm{NO_x}}$, REI_{CO} and REI_{HC} for $\mathrm{NO_x}$, CO and HC which then can be used to determine the emission indices as follow:

$$EI_{\mathrm{NO_x}} = REI_{\mathrm{NO_x}}\ e^{SH_c}\sqrt{\left(\frac{\delta_{amb}^{1.02}}{\theta_{amb}^{3.3}}\right)} \tag{2.37}$$

$$EI_{\mathrm{CO}} = REI_{\mathrm{CO}}\left(\frac{\theta_{amb}^{3.3}}{\delta_{amb}^{1.02}}\right) \tag{2.38}$$

$$EI_{\mathrm{HC}} = REI_{\mathrm{HC}}\left(\frac{\theta_{amb}^{3.3}}{\delta_{amb}^{1.02}}\right) \tag{2.39}$$

Where SH_c is the humidity correction factor and can be calculated as follow:

$$SH_c = -19(\omega - 0.0063) \tag{2.40}$$

$$\omega = \frac{0.62198(\phi)\mathrm{P_v}}{\mathrm{P} - (\phi)\mathrm{P_v}} \tag{2.41}$$

Where, ω is the specific humidity, ϕ is the relative humidity, and $\mathrm{P_v}$ is the saturation vapor pressure. The $\mathrm{P_v}$ can be defined as follow:

$$\mathrm{P_v} = (0.014504)10^{\beta} \tag{2.42}$$

Where the β is calculated as follow:

$$\begin{aligned}\beta = {} & 7.90298\left(1-\frac{373.16}{\mathrm{T}}\right)+3.00571+(5.02808)\log\left(\frac{373.16}{\mathrm{T}}\right)+ \\ & \left((1.3816)10^{-7}\right)\left[1-10^{11.344\left(1-\frac{\mathrm{T}}{373.16}\right)}\right]+\left((8.1328)10^{-3}\right)\left[10^{3.49149\left(1-\frac{373.16}{\mathrm{T}}\right)}-1\right]\end{aligned} \tag{2.43}$$

By using the emission indices of $\mathrm{NO_x}$, CO and HC from Eqs. (2.37 – 2.39) and fuel burn FB, the emissions of these gases can be defined as follow:

$$E_{\mathrm{NO_x}} = EI_{\mathrm{NO_x}}\ FB \tag{2.44}$$

$$E_{CO} = EI_{CO}\ FB \tag{2.45}$$

$$E_{HC} = EI_{HC}\ FB \tag{2.46}$$

Where, E_{NO_x}, E_{CO} and E_{HC} are respectively the emissions of NO_x, CO and HC in grams.

2.2 Performance Index

Already having the mathematical model of the system, the second key element to describe the aircraft trajectory optimization problem is the Performance Index (PI). A PI is a function, which defines the system's physical requirements into mathematical terms. When the PI is optimized, it indicates that the system is performing in the most desirable manner.

The PI has various forms depending on the measure of performance that is being optimized based on mission requirements. However, the general form of PI can be described as a problem of Bolza [68], [69], which can be defined as follows:

$$J = \Phi\left[t_f, X(t_f)\right] + \int_{t_0}^{t_f} L\left[t, X(t), U(t)\right] dt \tag{2.47}$$

The problem of Bolza Eq. (2.47) consists of two terms, the first term Φ referred to as the Mayer term, and the second term L referred to as the Lagrange term. The PI in Eq. (2.47) may contain just the Mayer term or the Lagrange term or both, depend on the mission requirements. The Mayer and Lagrange term are described below:

- *Mayer Term:* The Mayer term is a type of PI which represents the cost related to the terminal states and is a function of state variables. One example of this type of PI is terminal state problems, where the objective is to minimize the deviation of the final state of a system from its desired state. The PI for the terminal state problem can be defined as:

$$J = \left\| X(t_f) - r(t_f) \right\|^2 \tag{2.48}$$

  Where, $X(t_f)$ is the final state of a system and $r(t_f)$ is the desired final state.

- *Lagrange Term*: The Lagrange term represents the cost of the entire time history. These costs are integrals of control and state variables. One example of PI that uses the Lagrange term is the Direct Operating Cost (DOC) of flight, where the objective is to minimize the cost of the consumed fuel plus the other cost related to flight time. The PI for the direct operating cost of a flight can be defined as:

$$J = \int_{t_0}^{t_f} \left(FF + CI \right) dt \tag{2.49}$$

Where, t_0 and t_f are respectively the initial and final time, FF is the fuel flow rate of the aircraft, and CI is the cost index [70]. it is an adjustable constant parameter that represents the cost associated with fuel burn and flight time. For all aircraft models, the minimum value (Zero) of cost index results in maximum range airspeed and minimum trip fuel but ignores the time-related cost. When the cost index is maximum, it results in minimum flight time but ignores the fuel cost.

2.3 Boundary and Path constraints

Once, the mathematical model and the performance index (objective function) have been defined for the aircraft trajectory optimization problem, the next step is to define the Boundary and path constraints of the state and control variables. Generally, the inequality constraints that are imposed on the state and control variables for aircraft trajectory optimization problems are due to aircraft operational limitations (e.g., aerodynamic, structural, and propulsive, etc.). some examples of operational boundary and path constraints are as follow:

$$V_{\min} \le V(t) \le V_{\max} \qquad h_{\min} \le h(t) \le h_{\max}$$

$$\gamma_{\min} \le \gamma(t) \le \gamma_{\max} \qquad \psi_{\min} \le \psi(t) \le \psi_{\max}$$

$$T_{\min} \le T(t) \le T_{\max} \qquad \alpha_{\min} \le \alpha(t) \le \alpha_{\max}$$

$$\mu_{\min} \le \mu(t) \le \mu_{\max} \qquad m_{\min} \le m(t) \le m_{\max}$$

$$\left[x(t_0), y(t_0), h(t_0) \right] = \left[x_0, y_0, h_0 \right] \qquad \left[x(t_f), y(t_f), h(t_f) \right] = \left[x_f, y_f, h_f \right]$$

Where, (x_0, y_0, h_0) are the initial states, and (x_f, y_f, h_f) are the final states.

2.4 General Formulation of Trajectory Optimization Problem

Once, the key steps (i.e., mathematical modeling, performance index, and the boundary and path constraints of the system) to formulate the aircraft trajectory optimization have been defined. The problem now can be introduced as a general representation [2], [3], [71], [72], [73].

Considering a nonlinear system whose dynamics is modeled by a set of ordinary differential equations:

$$\dot{X}(t) = \mathrm{f}\left[t, X(t), U(t) \right] \tag{2.50}$$

With $t \in \mathbb{R}$ is the time (time $t \in [t_0, t_f]$ is the independent variable), $X(t) \in \mathbb{R}^n$ is the state vector, $U(t) \in \Omega \subset \mathbb{R}^m$ is the control vector, Ω a compact domain of feasible controls, $f : \mathbb{R} \times \mathbb{R}^n \times \Omega$, a vector-valued function, both the state and control are dependent on t .

The general TOP is to find an admissible control U^*, and the corresponding admissible state trajectory X^* that optimizes the Performance Index (PI):

$$J = \Phi\left[t_f, X(t_f)\right] + \int_{t_0}^{t_f} L\left[t, X(t), U(t)\right] dt \tag{2.51}$$

U^* is called an optimal control and X^* an optimal trajectory. This TOP may be subject to the equality c_{eq} , and the inequality c_{inq} constraints on the state and the control along the trajectory:

$$c_{eq}\left[X(t), U(t)\right] = 0 \tag{2.52}$$

$$c_{inq}\left[X(t), U(t)\right] \le 0 \tag{2.53}$$

It may also be subject to nonlinear boundary conditions Ψ , which enforces restrictions on the initial and final states of the system:

$$\Psi_{\min} \le \Psi\left[t_0, X(t_0), t_f, X(t_f)\right] \le \Psi_{\max} \tag{2.54}$$

The Eqs. (2.50) – (2.54) describe a single-phase TOP. However, in some applications the dynamics of the vehicle change over time. In these cases, TOP is typically divided into a collection of N phases in order to accommodate changes in the dynamics, where the PI can be defined as follows:

$$J = \sum_{k=1}^{N} J^{(k)} \tag{2.55}$$

The PI in each phase, $J^{(k)}$, $(k = 1, ..., N)$ can be subject to different system dynamics:

$$\dot{X}^{(k)}(t) = \mathrm{f}\left[t, X^{(k)}(t), U^{(k)}(t)\right] \tag{2.56}$$

Path constraints,

$$c_{eq}^{(k)}\left[X^{(k)}(t), U^{(k)}(t)\right] = 0 \tag{2.57}$$

$$c_{inq}^{(k)}\left[X^{(k)}(t), U^{(k)}(t)\right] \le 0 \tag{2.58}$$

And boundary conditions,

$$\Psi_{\min}^{(k)} \le \Psi^{(k)}\left[t_0, X^{(k)}(t_0), t_f, X^{(k)}(t_f)\right] \le \Psi_{\max}^{(k)} \tag{2.59}$$

Typically, the multi-phase trajectory is continuous, e.g., $X^{(k)}(t_f) = X^{(k+1)}(t_0)$. However, it may not always be the case. Eventually, all phases can be linked depending on the dynamic system behavior.

2.5 Summary

In this chapter, the formulation of the aircraft Trajectory Optimization Problem (TOP) is presented. The formulation process is divided into three key steps of mathematical modeling, defining the performance index, and defining the boundary and path constraints of the problem. In the mathematical model section, the aircraft dynamics, performance, and emissions models are presented in detail. In the performance index and boundary and path constraints sections, several examples of them are provided. Finally, the general formulation of TOP is presented.

Chapter 3

3. Numerical Approaches for Trajectory Optimization Problem

This chapter provides a brief review of the major numerical approaches that are used to solve the Trajectory Optimization Problem (TOP). In specific three basic classes of (e.g., indirect, direct, and dynamic programming) methods are discussed. The characteristics, advantages, and disadvantages of these methods are also discussed.

There are two types of approaches (e.g., analytical, and numerical) that exist to Solve TOPs. Although analytical approaches are the most desirable ones as they produce analytical solutions, they are not feasible for most practical cases. Typically, the TOPs are solved by a wide range of numerical approaches. They can be classified into three basic classes: indirect methods, direct methods, and dynamic programming.

In the indirect methods, the TOP is solved by the Pontryagin's Maximum Principle (PMP), where the original problem is transformed into Boundary Value Problem (BVP) by formulating the first-order necessary condition for optimality which derived from PMP [74], [75], [76]. The main advantages of indirect methods are that they lead to high accuracy solutions and guarantee that the solution satisfies the optimality condition. However, a good initial approximation of the co-state equation in indirect methods is required in order to converge, which is quite difficult to guess as the physical meaning of the co-state equation is not well established [77]. Besides, the BVPs that arise for many practical TOPs in indirect methods, are quite difficult to solve, because of the complex dynamics and constraints structure of the problem. It also demands computationally intensive iterative numerical procedures. Generally, the TOP results in a Two-Point Boundary Value Problem (TPBVP) for a single-phase problem and a Multi-Point Boundary Value Problem (MPBVP) for a multiple-phase problem.

The second basic class approach is the direct methods, which are based on the transformation of the original TOP into a Nonlinear Programming (NLP) problem, it is done by discretizing the state and control of an infinite-dimensional problem into a finite-dimensional problem [78]. The resulting NLP is then solved numerically by the well-established optimization techniques [79]. Although the indirect methods produce more accurate solutions than the direct methods, direct methods are easier to implement, and they tend to have better convergence properties over indirect methods. Another great advantage of direct methods is that they do not need an initial guess of co-states, as it is difficult to predict the time histories. The parameterization techniques have an important role in the convergence and accuracy of the solution. Generally, the direct methods vary for the variables to be parameterized (i.e., state and control), Such as, in the direct

shooting and direct multiple shooting techniques only the control is parameterized, differential inclusion parameterizes only the states, and in the direct collocation technique, both the states and controls are parameterized.

Aside from the first two numerical approaches (i.e., direct, and indirect methods), Dynamic Programming (DP) is another well-established method. The DP is based on a simple intuitive concept called the Principle of Optimality (PO), which is used a necessary condition of optimality to solve the TOP, it has the appealing feature of splitting the complex optimization problem into a sequence of simple optimization subproblems, and the solution obtained by the DP is guaranteed to be absolute (global) optimum [18]. The numerical framework of DP is also very suitable to handle discrete-time dynamic systems with nonlinear characteristics [80]. Another great advantage of the DP is that it is ideally suited for digital computers, as the optimization procedure is carried out stage by stage calculations. However, the computational space and time complexity demanded by the DP is enormous due to the immense number of grid points required to find the optimum, coupled with the interpolation problem (when the trajectory from a grid point does not reach exactly to the next grid point), prevent the use of the DP in many practical real-time implementations especially with high dimension problems.

Fig. (3.1) represents various numerical methods that fall into three basic categories to solve the TOP. Each of these three categories and the most common techniques that fall under them are described in detail in the next subsections.

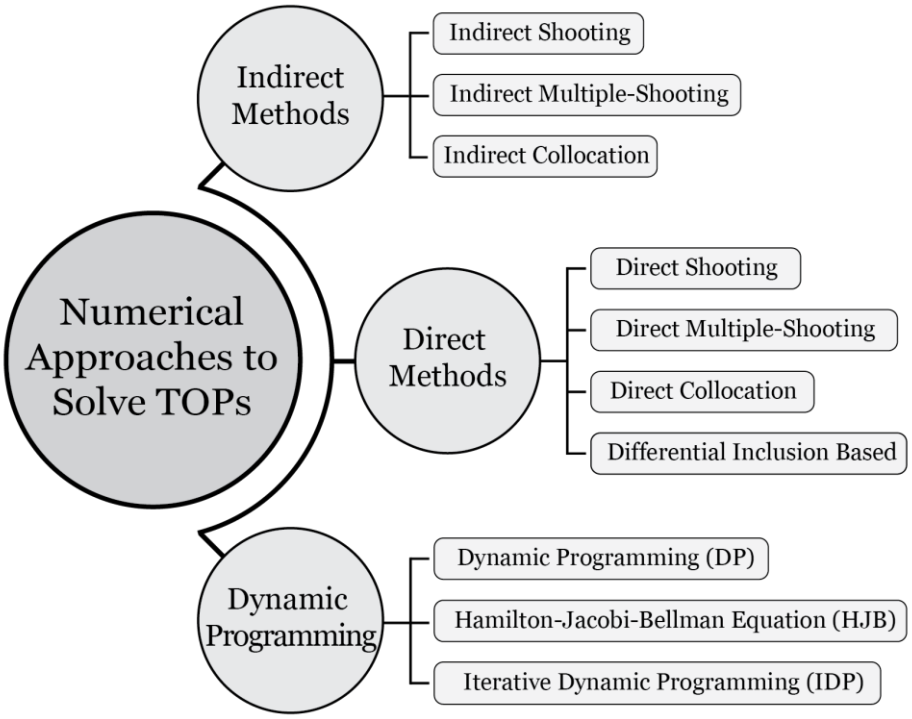


Figure 3.1: Numerical approaches to solve trajectory optimization problem.

3.1 Indirect Methods

The Trajectory Optimization Problem (TOP) is solved by the Calculus of Variation (CoV) in the indirect method, where the original TOP is converted into a Boundary Value Problem (BVP) by formulating the first-order necessary condition for optimality which is derived from Pontryagin's Maximum Principle (PMP).

Considering a nonlinear system whose dynamics is modeled by the state equation:

$$\dot{X}(t) = \mathrm{f}\left[t, X(t), U(t)\right] \tag{3.1}$$

and the Performance Index (PI) that need to be optimized is:

$$J = \Phi\left[t_f, X(t_f)\right] + \int_{t_0}^{t_f} L\left[t, X(t), U(t)\right] dt \tag{3.2}$$

Now, introducing the Hamiltonian function as follows [71], [72], [73]:

$$H = L\left[t, X(t), U(t)\right] + \lambda_c^{\,T}(t)\mathrm{f}\left[t, X(t), U(t)\right] \tag{3.3}$$

Where, H is the Hamiltonian, $\lambda_c(t) \in \mathbb{R}^n$ is the co-state or adjoint, $X(t) \in \mathbb{R}^n$ is the state, and $U(t) \in \mathbb{R}^m$ is the control. For a simple single-phase TOP without any static parameters, the first-order necessary conditions of optimality (also known as Euler-Lagrange equations) of the continuous-time problem are given as follows:

State equation,

$$\dot{X}(t) = \mathrm{f}[t, X(t), U(t)] = \left[\frac{\partial H}{\partial \lambda}\right]^T \tag{3.4}$$

Co-state equation,

$$\dot{\lambda}_c(t) = -\left[\frac{\partial H}{\partial X}\right]^T \tag{3.5}$$

The Eqs. (3.4) and (3.5) are also called a Hamiltonian system, and the control equation is,

$$\frac{\partial H}{\partial U} = 0 \tag{3.6}$$

The control Eq. (3.6) can be presented by a more general expression of the equation, which is known as PMP,

$$U^* = \arg\min_{U \in \Omega} H \tag{3.7}$$

The PMP states that the control variable must be selected to optimize the Hamiltonian subject to state and control path constraints. As Eq. (3.7) is minimizing the Hamiltonian is also called a

Pontryagin's minimum principle in order to be consistent with the algebraic sign conventions [74], [75], [76].

The final boundary condition of the problem is:

$$\left[H+\frac{\partial \Phi}{\partial t}\right]_{t_f} \delta t_f + \left[\frac{\partial \Phi}{\partial X} - \lambda_c(t)\right]_{t_f} \delta X_f = 0 \tag{3.8}$$

Where, Φ is Mayer term of the PI. Now, the transversality conditions depend on the terminal time and terminal states as follow:

In case, the terminal time t_f is fixed, the terminal state $X(t_f)$ is free,

$$\lambda_c(t_f) = \left[\frac{\partial \Phi}{\partial X}\right]_{t_f} \tag{3.9}$$

In case, the terminal time t_f is free, the terminal state $X(t_f)$ is fixed,

$$H(t_f) + \left[\frac{\partial \Phi}{\partial t}\right]_{t_f} = 0 \tag{3.10}$$

The Eqs. (3.4) – (3.10) provide the necessary conditions of optimality which is often referred to as a Two-Point Boundary Value Problem (TPBVP), that can be solved numerically to obtain the candidate solution or extremal of the TOP. The advantages that the indirect methods provide are as follow:

- Indirect methods produce high-accuracy solutions and guarantee that the solution satisfies the optimality condition.
- The optimal control of the problem is determined analytically.
- The co-states variables of the indirect method represent the sensitivity of cost to changes in their corresponding state variables.

The disadvantages of indirect methods are as follow:

- Indirect methods yield a TPBVP, which is difficult to solve in many practical TOP because of the complex dynamics and constraints structure of the problem.
- A good initial approximation of the co-state equation is required in order to converge, which is difficult to guess as the physical meaning of the co-state equation is not well established.
- Changes in the problem formulation, (i.e., modification of the system equations), require formulating once more the necessary condition of optimality of the problem.

Presently, the most common numerical techniques for solving the TPBVP are indirect Shooting, indirect multiple shooting, indirect collocation as illustrated in Fig. (3.1). Each of these techniques are presented in the next subsections.

3.1.1 Indirect Shooting

The simplest indirect method to solve the TOP is the indirect shooting method. In this method, the entire trajectory is approximated by a single segment. Where, it requires an initial guess of the unknown boundary conditions, and some boundary conditions of the trajectory are generally known, which is then used to integrate the Hamiltonian system Eqs. (3.4) and (3.5) from t_0 to t_f . The terminal conditions from the numerical integration are obtained by reaching the terminal time t_f, which is then compared with the known terminal conditions. The difference between the terminal conditions is known as defect constraint Δ. If the defect constraint has a value more than a specified tolerance ε, the process is then repeated by adjusting the initial guess of the unknown boundary conditions until the defect constraint is within the specified tolerance close to zero.

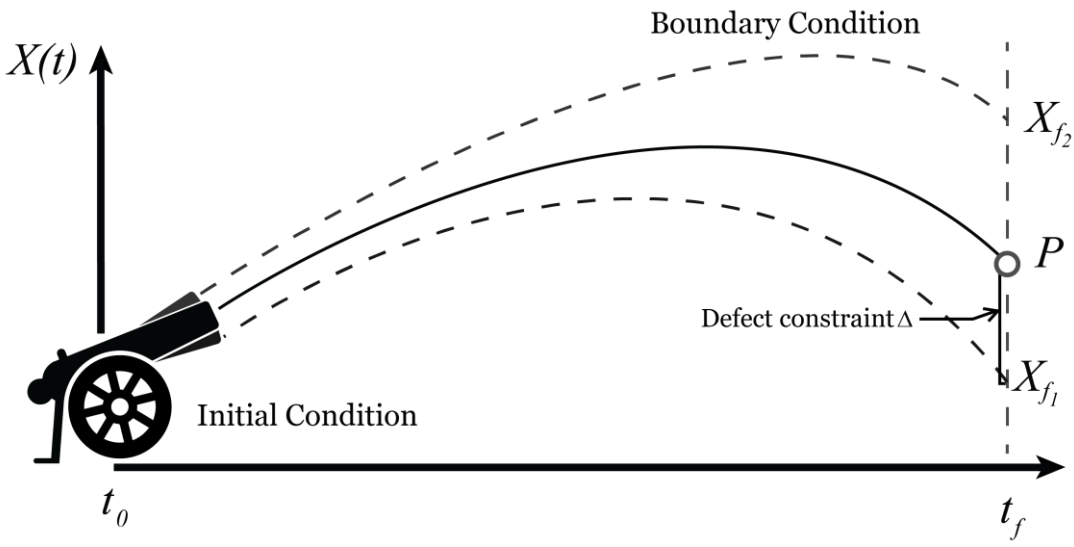


Figure 3.2: Diagram of indirect shooting method.

Fig. (3.2) shows a well-known simple example of an indirect shooting method. Where a cannon is aimed to hit a known target P and the initial state and terminal state of the trajectory are known. The objective is to find the initial angle of the cannon, so the cannonball hits the known target.

Indirect shooting approximates the trajectory by using the initial guess of the cannon angle. In other words, the cannon is fired with the initial angle that was chosen, if the cannonball does not hit the target, the angle is adjusted to reduce the defect constraints. The process is repeated until the defect constraint is within the specified tolerance close to zero.

The main advantage of indirect shooting is that it deals with a small number of variables. The main drawback of the method is that a small change in the initial condition can produce a very large change in the final condition [76].

3.1.2 Indirect Multiple-Shooting

The indirect multiple-shooting method is an extension of the simple indirect shooting method. The most significant disadvantage of the shooting method is that small changes in the initial guess of the unknown boundary condition of the trajectory can propagate into very nonlinear changes at the final state of the trajectory. In order to overcome this disadvantage of the simple shooting method, one approach is to simply break the problem into multiple segments, called the multiple-shooting method [78], [81], [82].

In this method, the time domain is divided into $(M+1)$ subintervals of the form $t_0 < t_1 < t_2 \cdots < t_M < t_f$. For each interval $[t_i, t_{i+1}]$ where $i = 0, \cdots, M$ the shooting method is then applied with the initial values of the state and the initial guess of the unknown costate.

To ensure the continuity between consecutive intervals of the trajectory the following constraints need to be enforced, which are also called defect constraints.

$$Y_i^- = Y_i^+ \tag{3.11}$$

Where, $Y(t_i^-) = Y_i^-$, $Y(t_i^+) = Y_i^+$ and $Y(t)$ is the combined state X and costate vector Y, i.e.

$$Y(t) = \begin{bmatrix} X(t) \\ \lambda_c(t) \end{bmatrix} \tag{3.12}$$

The defect constraints of Eq. (3.11) results in a vector root-finding problem, where it is desired to drive the defect constraint to zero $(Y_i^- - Y_i^+) = 0$. Fig. (3.3) illustrates the diagram of an indirect multiple-shooting method.

Compare to the single shooting method the multiple shooting method requires additional variables and constraints for each segment which increases the size of the problem to be solved. Despite this drawback, the multiple shooting is an improvement over the single shooting method as this method enhances the robustness. This method also reduces the sensitivity to errors in the unknown initial conditions as the integration is performed over significantly smaller time intervals.

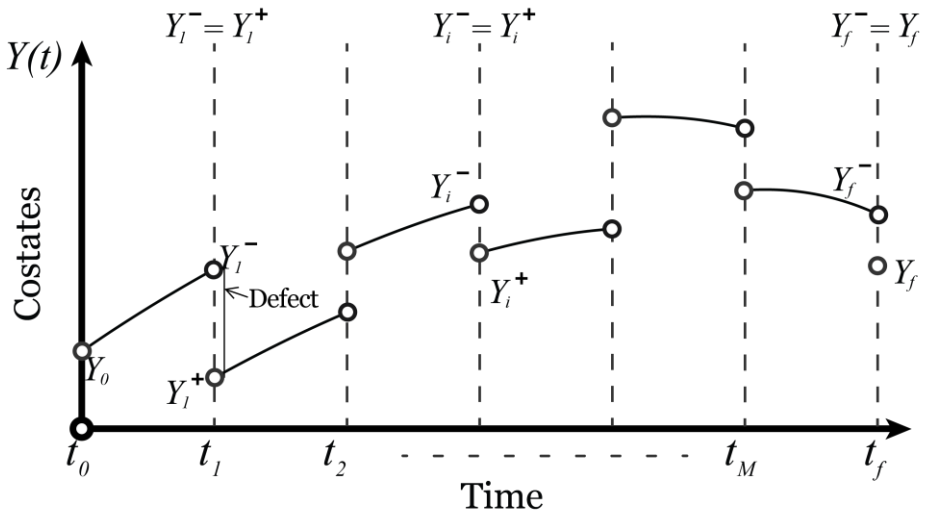


Figure 3.3: Diagram of indirect multiple-shooting method.

3.1.3 Indirect Collocation

The multiple boundary value problem, that arises from the trajectory optimization problem can be effectively solved by the indirect collocation technique, which makes it one of the most popular indirect methods. This method came across in the aerospace community by Dickmanns and well in the 1980s, they used the collocation scheme to solve the Two-Point Boundary Value Problem (TPBVP) of the indirect method [83], [84].

In this technique, both the state and co-state are parameterized using piecewise polynomials which can be easily integrated and differentiated. This leads to a root-finding problem. Then the system of nonlinear equations can be solved using appropriate root-finding techniques. This indirect collocation method eliminates the sequential nature of the shooting method which makes it more robust.

3.2 Direct Methods

Unlike the indirect methods, the direct methods do not require deriving the necessary condition of optimality to solve the Trajectory Optimization Problem (TOP). The main idea is to transform the original TOP from an infinite-dimensional problem into a finite-dimensional Nonlinear Programming (NLP) problem [78], [85], [86]. Thus, the direct methods follow two steps to solve the TOP problem: first they discretize the original problem by using numerical techniques then they solve the resulting NLP by two different categories of optimization methods as follow [3]:

Gradient-based methods: The most common gradient-based methods to solve NLP are Sequential Quadratic Programming (SQP), and Interior-Point (IP) methods. As these methods use gradient information, they have well-defined convergence criteria. This has led to extensive research on SQP and IP methods and the development of many software programs for the

numerical solution of NLPs, such as NPSOL [87], SNOPT [88], KNITRO [89], IPOPT [90], rSQP++ [91], etc. Other than these NLP solvers another well-known NLP solver is *fmincon()* of MATLAB optimization toolbox.

Metaheuristics methods: Recent years, metaheuristic methods are becoming popular to solve the NLP. These methods find the optimum by iteratively acting to improve candidate solutions in a stochastic manner. The most common metaheuristic methods are [92]:

- Genetic algorithm (GA) is an evolutionary approach based on the principle of "survival of the fittest" [93], [94].
- Simulated annealing (SA) is inspired by the physical process of annealing in metallurgy [95].
- Particle swarm optimization (PSO) is a population-based stochastic optimization method, based on the idea of swarms of animals [96], [97].
- Ant colony optimization (ACO) is inspired by the behavior of ants of an ant colony [98].

Although the direct methods are less accurate than the indirect methods, yet they are far more popular to solve practical TOP because of the following advantages:

- Direct methods are easier to implement and do not require formulating the necessary condition of optimality to solve the TOP.
- They tend to have better convergence properties over other methods.
- More robust, they do not need an initial guess of co-states, as it is difficult to predict the time histories.

The disadvantages of the direct methods are as follow:

- Direct methods require an initial guess of the solution for the NLP solver to find an improved solution.
- Generally, the states and controls are known only at discrete points.
- These methods are Prone to converge to a local minimum in the neighborhood of the initial guess.

The most common numerical techniques to convert the infinite-dimensional TOP to finite-dimensional NLP are direct Shooting, direct multiple shooting, direct collocation, and differential inclusion based technique. These techniques are described in the next sub-sections.

3.2.1 Direct Shooting

The direct shooting method [99], [100] is the most basic direct method to solve the Trajectory Optimization Problem (TOP). This method has the ability to describe the TOP in terms of a relatively small number of optimization variables.

Direct shooting is a control parameterization method. The control variables are discretized in the time domain, then the state variables are obtained using numerical integration as a function of control parameters to resolve the performance index (PI) Eq. (3.2). The control is parameterized by a specified function form as:

$$U(t) \approx \sum_{i=1}^{m} a_i Kn_i(t) \tag{3.13}$$

Where $Kn_i(t)$, $(i = 1,...,m)$ are known functions and a_i, $(i = 1,...,m)$ are the parameters to be determined from optimization.

The main advantage of the direct shooting method is that unlike direct collocation and Pseudospectral methods, which discretize both the state and control variables, this method only discretizes the control variables. In other words, the direct shooting method reduces the searching space by half than the direct collocation and Pseudospectral methods. The single shooting method is efficient for problems that are either simple or have an extremely good initialization. This method is widely used for launch vehicle trajectory and orbit transfer optimization applications mainly because this class of problem requires a relatively small number of Nonlinear Programming (NLP) variables.

3.2.2 Direct Multiple-Shooting

Direct multiple-shooting was introduced in the 1980s by Bock and Plitt [101]. Similar to the indirect multiple-shooting method, in a direct multiple-shooting method, in order to minimize the sensitivity error present in a simple shooting method, rather than representing the entire trajectory as a single segment, the trajectory is divided into multiple smaller segments. Thus, the time domain is divided into $(M+1)$ subintervals of the form $t_0 < t_1 < t_2 \cdots < t_M < t_f$.

For each interval $[t_i, t_{i+1}]$ where $i = 0, \cdots, M$ the direct shooting method is then applied as described in the previous subsection. To ensure the continuity between consecutive intervals of the trajectory the following constraints are enforced at the end of each subinterval.

$$X_i^- = X_i^+ \tag{3.14}$$

Where, $X(t_i^-) = X_i^-$, $X(t_i^+) = X_i^+$ The Eq. (3.14) results in vector root-finding problem where it is desired to drive the defect constraint to zero $(X_i^- - X_i^+) = 0$.

Although the direct multiple-shooting method requires dealing with additional variables and constraints, still it performs better than the simple shooting method, as the method significantly improves the robustness and reduces the sensitivity to errors over the shooting method. Fig. (3.4) illustrates the Diagram of the direct multiple-shooting method.

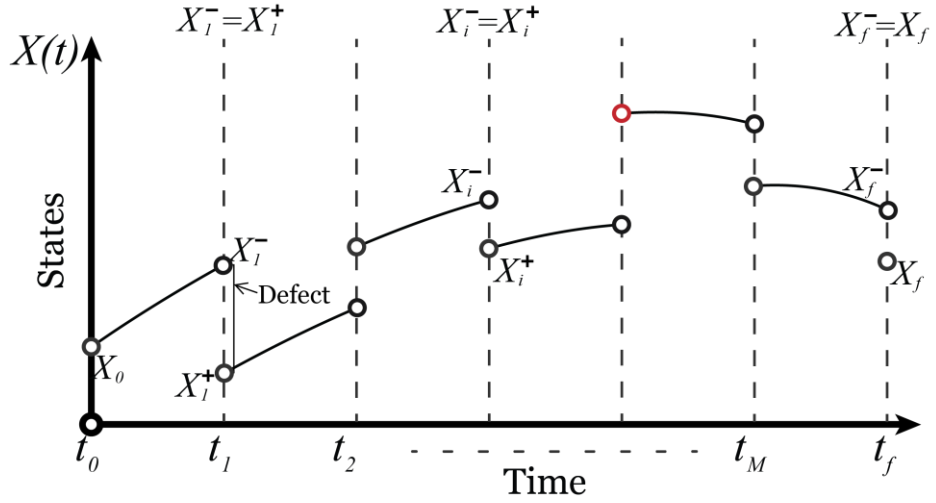


Figure 3.4: Diagram of direct multiple-shooting method.

3.2.3 Direct Collocation

The direct collocation method is one of the most popular methods to solve the Trajectory Optimization Problem (TOP). Unlike shooting methods in collocation methods both the state and control are parameterized. Thus, these methods are also known as the state and control parameterization methods [102]. Direct collocation methods involve the discretization of the differential dynamic constraints into a set of algebraic constraints using polynomial interpolation up to a certain degree with a grid of N points in the time interval $[t_0, t_f]$, $t_0 < t_1 < t_2 \cdots < t_N = t_f$. An interpolant (interpolating function) passes through the state values and maintains the state derivatives at the nodes. Then the interpolant is evaluated at points between the nodes called collocation points. At each collocation point, the interpolant derivatives are compared to the state derivatives, the difference is referred to as the defect. The defect is driven to zero in order to satisfy the equation of motion (EOM).

A basic form of direct collocation is Hermite-Simpson collocation [103], [104]. In this method, the problem is discretized by nodes and the midpoint or collocation point between the nodes is approximated by a third-order Hermite interpolating polynomials, which is illustrated in Fig. (3.5). In this method, the distribution of nodes is equally spaced in the time interval $[t_0, t_f]$, for each segment $[t_i, t_{i+1}]$, at the two nodes there are the states and their first derivatives are available $[X_i, X_{i+1}, \mathrm{f}_i, \mathrm{f}_{i+1}]$. The derivatives at the collocation point $\dot{X}_c$ in the segment $[t_i, t_{i+1}]$ are compared to the EOM, to ensure that the approximating polynomials accurately represent the EOM. The accuracy is achieved when the defect Δ is driven towards zero.

$$\Delta = \mathrm{f}_c - \dot{X}_c = \mathrm{f}\left(\mathrm{X}_c, \mathrm{U}_c, t_c\right) - \dot{X}_c = 0 \tag{3.15}$$

Recently, instead of using the third-order Hermite interpolating polynomials, some methods use N^{th} order polynomials, where the distribution of the collocation points between nodes is set by the Legendre-Gauss-Lobatto points. This approach is known as Hermite-Legendre-Gauss-Lobatto (HLGL) collocation method [105].

The Nonlinear Programming (NLP) problem that arises from implementing the direct collocation methods may be very large. Generally, the NLP has hundreds to tens of thousands of variables and constraints. Although having a large number of variables and constraints, these large NLP problems are easier to solve than Boundary Value Problems (BVPs), as the direct collocations methods often lead to sparse NLP and there are various NLP solvers (such as SNOPT [88], KNITRO [89]) to solve such problem efficiently.

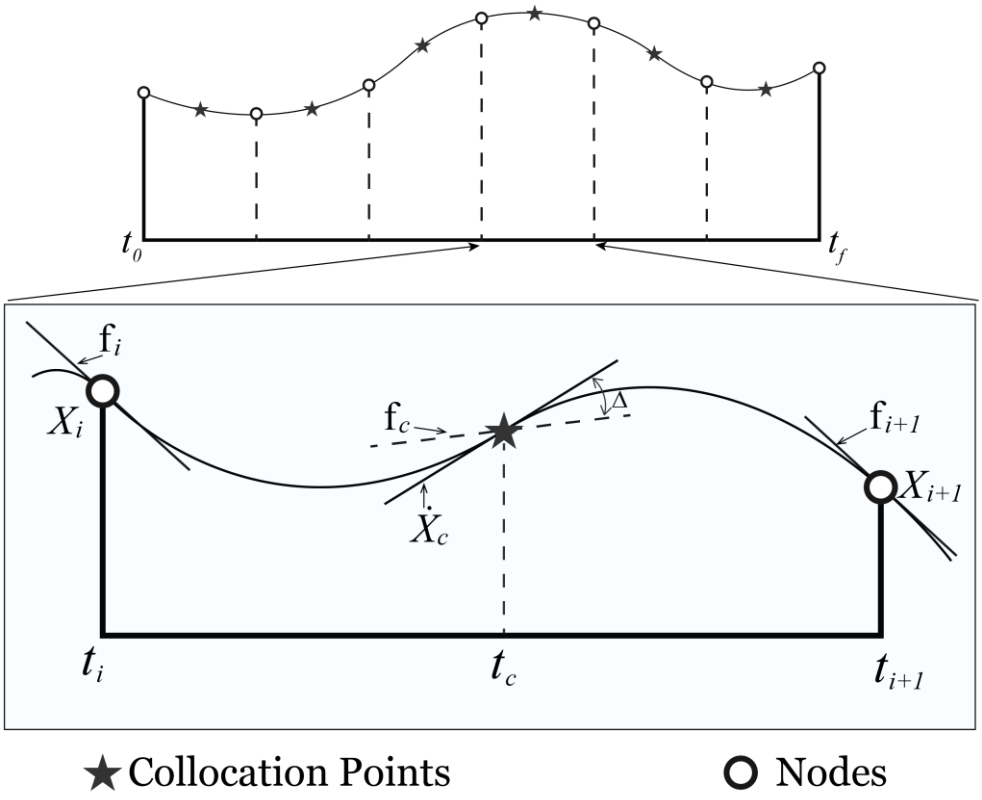


Figure 3.5: Diagram of Hermit-Simpson collocation method.

3.2.3.1 Pseudospectral Collocation

A pseudospectral (PS) collocation methods are a class of direct collocation methods also known as global orthogonal collocation methods [106], [107]. In a PS method, the state and control are approximated using a global interpolating polynomial at a set of nodes. The time derivative of the state in the dynamic equations is approximated by the first derivative of the interpolating polynomial, then at a set of collocation points, the interpolant derivatives are compared to the state derivatives, the difference between them is driven to zero in order to satisfy the EOM. PS collocation methods have become increasingly popular over the past several decades, generally because they have a faster convergence rate than other methods.

In recent years within the aerospace community several PS collocation methods have been developed where the most used PS collocation methods to solve the trajectory optimization problem (TOP) are [108]:

- Chebyshev Pseudospectral Method (CPM): The CPM uses the Chebyshev polynomials to approximate the state and control and perform orthogonal collocation at Chebyshev-Gauss-Lobatto (CGL) Points [109], [110], [111], [112].
- Legendre Pseudospectral Method (LPM): The LPM has employed Lagrange polynomials for the approximation of the state and control and uses Legendre-Gauss-Lobatto (LGL) Points for the orthogonal collocation [106], [113].
- Gauss Pseudospectral Method (GPM): In the GPM, the state and control are approximated using a basis of Lagrange polynomials same as LPM, but instead of LGL point the orthogonal collocation is performed at the Legendre-Gauss (LG) points [114], [115].
- Radau Pseudospectral Method (RPM): Identical to the LPM and GPM the RPM also uses the Lagrange polynomials as basis functions and the orthogonal collocation is performed at the Legendre-Gauss-Radau (LGR) points [116], [117].

3.2.4 Differential Inclusion Based

Differential Inclusion (DI) based method can be taken as the improvement of the collocation method, where the method is based on transforming the differential equations to differential inclusions [118], [119]. The DI based method enforces the Equation of Motion (EOM) at each node by applying inequality constraints on the state derivatives. The inequality constraints are obtained by substituting the upper and lower bounds on the controls into the EOM. Unlike the collocation methods, the DI based method reduces the problem size, as this method effectively eliminates the controls from the optimization process. This advantage makes this method ideal for the online and real-time implementation of trajectory optimization of flight vehicles. However, this method has the advantage of online and real-time implementation, only when the controls appear linearly in the system model, which makes it not suitable to be used widely.

3.3 Dynamic Programming

Dynamic Programming (DP) is a well-established numerical optimization method to solve the Trajectory Optimization Problem (TOP). DP was first developed by Richard E. Bellman in the 1950s, the DP centers on a simple intuitive concept called the Principle of Optimality (PO), the PO is used as a necessary condition of optimality to solve the TOP in DP, where it splits the global optimization problem into local optimization subproblems and explores all the feasible state candidates that satisfy the necessary condition [18], [120]. The PO stated by Bellman as follows:

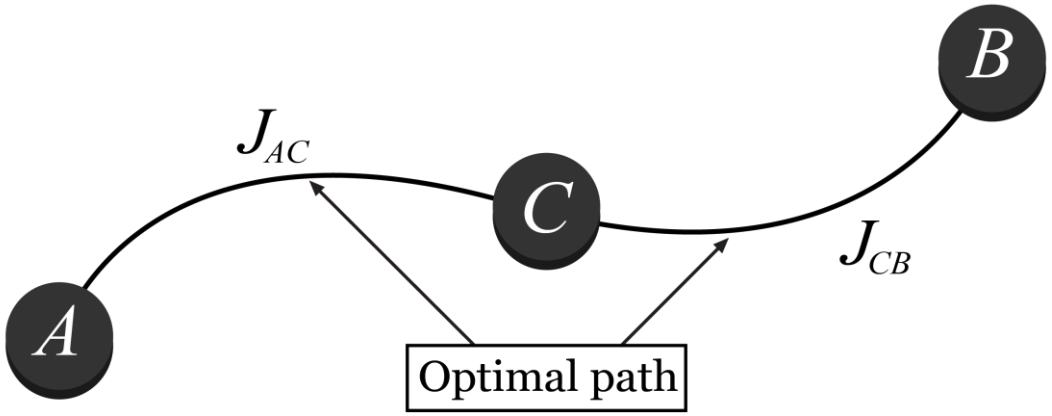


Figure 3.6: Diagram of Principle of Optimality.

"*An optimal policy has the property that whatever the initial state and the initial decision are (i.e., control), the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. (Richard E. Bellman)"*

Considering a simple multistage decision optimization process illustrated in Fig. (3.6). If the optimal path from a node A to B passes through a node C, then the PO assumes that the sub-path from A to C and C to B are also optimal. If the optimizing cost function for the sub-path A to C is J_{AC} and for sub-path C to B is J_{CB}, then the optimizing cost for the entire path A to B is:

$$J_{AB} = J_{AC} + J_{CB} \tag{3.16}$$

In other words, one can break the total optimal path into smaller segments that are themselves optimal. Conversely, if one finds the optimal values for these smaller segments, then one can obtain the optimal value for the entire path. Kirk presented a simple proof of contradiction of PO [80].

Generally, there are two different DP approaches to solve the TOPs [72]. The basic DP approach is to approximate the continuous operating systems by discrete operating systems, this approach leads to the recurrence equation of DP that is ideally suited for digital computers. The alternative approach is to apply the DP directly to continuous operating systems, which leads to a nonlinear partial differential equation otherwise known as the Hamilton-Jacobi-Bellman (HJB) equation. Both approaches are described in the next sub-sections.

3.3.1 Basic Approach

The basic approach to apply the numerical procedure of Dynamic Programming (DP) consists of approximating the system differential equations of a continuous system by the difference equations and approximating the integral in the Performance Index (PI) by a summation.

Considering a nonlinear system whose dynamics is modeled by a set of ordinary differential equations as in Eq. (3.1), which can be approximated by a set of difference equations as follow:

$$X_{k+1} = \mathrm{f}_d\left[t_k, X_k, U_k\right] \tag{3.17}$$

Where, X_k and U_k, $(k = 0,1,...,N-1)$ are respectively the state and control variables with appropriate boundary conditions. And the PI of Eq. (3.2) can be approximated by a summation as follow:

$$J = \Phi\left[t_N, X_N\right] + \sum_{k=0}^{N-1} L_d\left[t_k, X_k, U_k\right] \tag{3.18}$$

Generally, the optimization process of DP proceeds stage-wise starting with the terminal stage $k = N$ and continues recursively down to the initial stage $k = 0$. In the first stage of the optimization process when $k = N$, the PI of Eq. (3.18) only depends on the cost related to the terminal states (i.e., Mayer term Φ) as follow:

$$J_N^*\left(X_N\right) = \Phi\left[t_N, X_N\right] \tag{3.19}$$

Where, J_N^* is the optimal cost of reaching the terminal state X_N. Assuming, that the optimal control, state, and cost are known from any stage $k+1$ to the final stage N. Then at any stage, k the principle of optimality (PO) states that whatever the initial state and the initial decision, in this case, X_k and U_k, the remaining decision U_{k+1} must be optimal with regarding the state X_{k+1} that results from the first decision U_k. Thus, the PI can be rewritten as:

$$J_k^*\left(X_k\right) = \min_{U_k}\left[L_d\left[t_k, X_k, U_k\right] + J_{k+1}^*\left(X_{k+1}\right)\right] \tag{3.20}$$

The Eq. (3.20) is the mathematical form of PO, it is also known as the functional equation of DP [73]. Where, J_{k+1}^* represents the cost of the optimal path from any stage $k+1$ to the final stage N, and J_k^* is the optimal cost from any stage k to the final stage N.

Since the state X_{k+1} is related to the next state X_k and control U_k as in Eq. (3.17), the Eq. (3.20) can be rewritten as:

$$J_k^*(X_k) = \min_{U_k}\left[L_d\left[t_k, X_k, U_k\right] + J_{k+1}^*\left(\mathrm{f}_d\left[t_k, X_k, U_k\right]\right)\right] \tag{3.21}$$

Eq. (3.21) is the recurrence relation of DP. This recurrence equation is solved by finding the optimal control law or optimal policy U_k^*, $(k = 1, 2, ..., N)$, which is obtained by trying all admissible control values at each admissible state value of the system. Typically, the admissible state and control values are quantized into a finite number of levels in order to make the computational procedure feasible.

Fig. (3.7) illustrates the stagewise optimization procedure of DP, where two-dimensional state space has been considered, thus, the state vector contains 2 state variables $X = [x_1, x_2]$. The dotted points in the figure are the quantized state values at which all the quantized control values need to be applied.

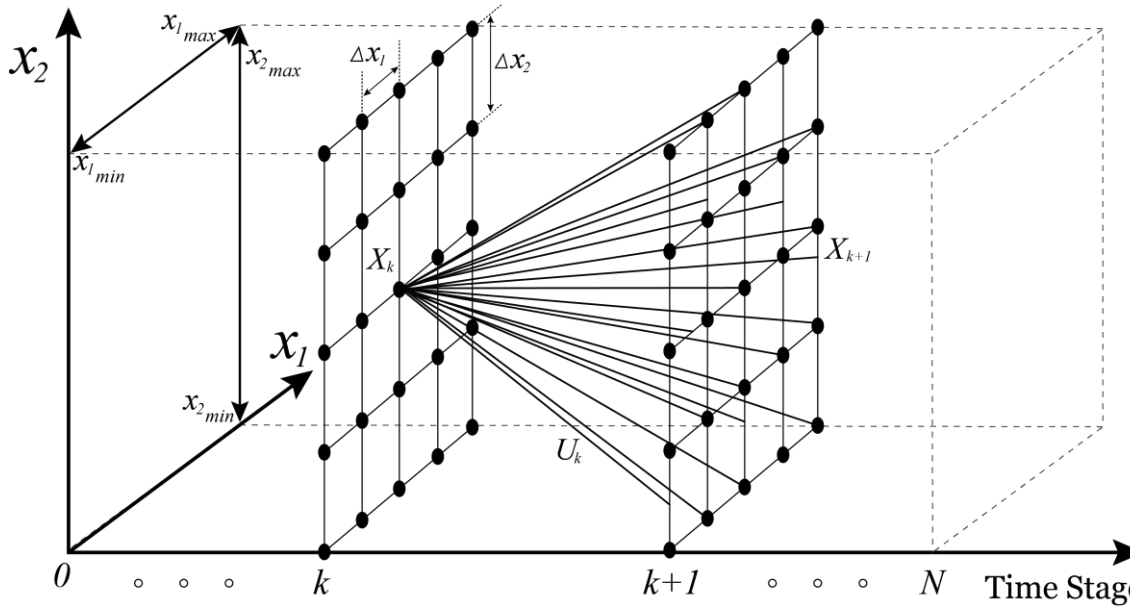


Figure 3.7: Stagewise optimization procedure of DP.

DP has many appealing features to solve the trajectory optimization problem (TOP) as follow:

- The solution obtained by the DP is guaranteed to be absolute (global) optimum, as the method uses direct search to solve the recurrence equation Eq. (3.21).
- The numerical framework of DP is ideally suited to handle equality or inequality constraints and nonlinear characteristics of the system.
- DP splits a complex optimization problem into a sequence of simple optimization subproblems, this stage-by-stage optimization procedure is ideally suited for digital computers.

However, there are serious limitations of the DP method, which prohibits the use of DP for high-dimensional systems as follow:

- *The curse of dimensionality:* This is the most serious drawback of DP, it requires an enormous amount of calculation for a reasonable higher-dimensional problem since a large number of grid points are required and then one has to go through all the combination of all grid points in the state and control domain.

- *The menace of the expending grid:* This is another drawback of DP, an interpolation problem arises, when the trial control values of the system do not drive the state from a grid point to a grid point at the next stage [121]. Fig. (3.8) illustrates the menace of the expending grid of DP, where the state values that did not reach the next grid points are shown in grey. Generally, by increasing the grid points this problem can be solved however, it also increases the storage requirement and the computational time.

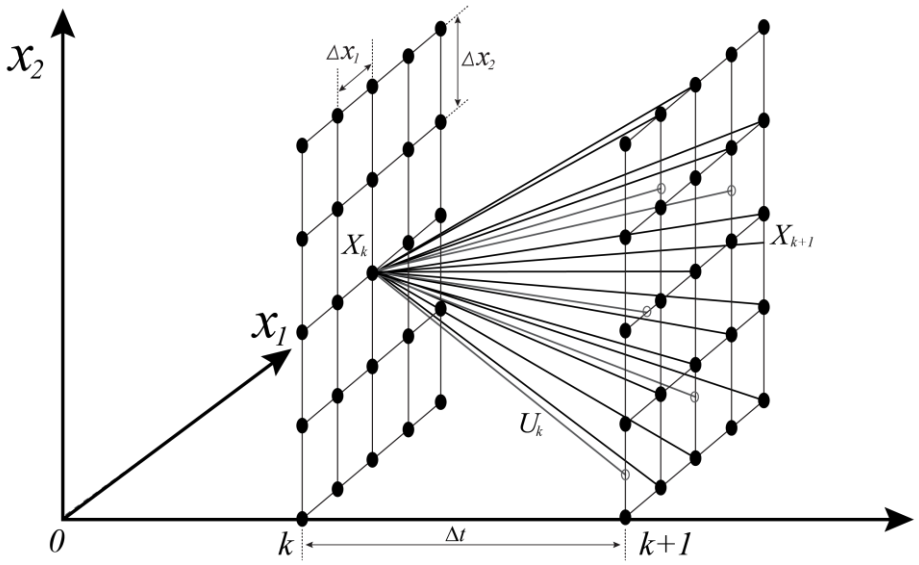


Figure 3.8: The menace of the expending grid of DP.

Although the DP guarantees global optimality, and it can easily handle nonlinear characteristics, equality, and inequality constraints. The requirement of enormous storage and computational time has prohibited the use of DP in many practical higher-dimension problems. These drawbacks of DP have limited its further development and acceptance in the aerospace field from the 1970s onward. However, the rapid increase of capability in recent computers with high-performance processors and parallel processing technology, have encouraged the revival of DP in many practical engineering problems.

3.3.2 Hamilton-Jacobi-Bellman Approach

In this section, an alternative approach of Dynamic Programming (DP) which can be applied directly to the Continues operating system is presented, which leads to a nonlinear partial differential equation otherwise known as the Hamilton-Jacobi-Bellman (HJB) equation.

Considering the state equation and Performance Index (PI) as in Eqs. (3.1) and (3.2). The minimum value of PI for an initial state $X(t)$ at a time t can be defined as follow:

$$J^*\left(t, X(t)\right) = \min_{U(t)} \left\{ \int_t^{t_f} L\left[t, X(t), U(t)\right] dt + \Phi\left[t_f, X(t_f)\right] \right\} \tag{3.22}$$

The $J^*\left(t, X(t)\right)$ is the PI when evaluated along the optimal trajectory from time t to terminal time t_f, where $\left[t \le t_f\right]$. Now by subdividing the interval, the Eq. (3.22) becomes:

$$J^*\left(t, X(t)\right) = \min_{U(t)} \left\{ \int_t^{t+\Delta t} L\left[t, X(t), U(t)\right] dt + \int_{t+\Delta t}^{t_f} L\left[t, X(t), U(t)\right] dt + \Phi\left[t_f, X(t_f)\right] \right\} \tag{3.23}$$

From Eqs. (3.22) and (3.23) $J^*\left(t, X(t)\right)$ can be rewritten as:

$$J^*\left(t, X(t)\right) = \min_{U(t)} \left\{ \int_t^{t+\Delta t} L\left[t, X(t), U(t)\right] dt + J^*\left(t + \Delta t, X(t + \Delta t)\right) \right\} \tag{3.24}$$

Where, $J^*\left(t + \Delta t, X(t + \Delta t)\right)$ is the minimum cost for an initial state $X(t + \Delta t)$ from time $t + \Delta t$ to terminal time t_f. For small Δt and by using the Taylor series, the Eq. (3.24) becomes:

$$J^*\left(t, X(t)\right) = \min_{U(t)} \begin{Bmatrix} L\left[t, X(t), U(t)\right]\Delta t + J^*\left(t, X(t)\right) + J_t^*\left(t, X(t)\right)\Delta t \\ + J_X^{*T}\left(t, X(t)\right) \mathrm{f}\left[t, X(t), U(t)\right]\Delta t + o(\Delta t) \end{Bmatrix} \tag{3.25}$$

Where, $o(\Delta t)$ is the term that contains higher orders Δt that arise from the approximation of the integral and the truncation of the Tylor series expansion, $J_t^* = \dfrac{\partial J^*}{\partial t}$, and $J_X^* = \dfrac{\partial J^*}{\partial X}$. Next removing the terms $J^*\left(t, X(t)\right)$ and $J_t^*\left(t, X(t)\right)$ from the minimization, in Eq. (3.25) as the minimization only depends on control $U(t)$, and dividing it by Δt and assuming $\Delta t \longrightarrow 0$, gives:

$$0 = J_t^*\left(t, X(t)\right) + \min_{U(t)} \left\{ L\left[t, X(t), U(t)\right] + J_X^{*T}\left(t, X(t)\right) \mathrm{f}\left[t, X(t), U(t)\right] \right\} \tag{3.26}$$

The boundary condition for this partial differential equation is:

$$J^*\left(t_f, X(t_f)\right) = \Phi\left[t_f, X(t_f)\right] \tag{3.27}$$

Now, the Hamiltonian function is:

$$H\left[t, X(t), U(t), J_X^*\right] = L\left[t, X(t), U(t)\right] + J_X^{*T}\left(t, X(t)\right) \mathrm{f}\left[t, X(t), U(t)\right] \tag{3.28}$$

Now comparing the Hamiltonian function in Eqs. (3.3) and (3.28), the co-state can be defined as follow:

$$\lambda(t) = \left(\frac{\partial J^*}{\partial X} \right) = J_X^* \tag{3.29}$$

Next using the Eqs. (3.26) and (3.28), it gives:

$$0 = J_t^*\left(t, X(t)\right) + H\left[t, X(t), U^*(t), J_X^*\right] \tag{3.30}$$

This equation Eq. (3.30) is the continuous-time analog of recurrence equations of DP Eq. (3.21), it is also known as the HJB equation. Where, $H\left[t, X(t), U^*(t), J_X^*\right]$ is the minimum of Hamiltonian function Eq. (3.28).

The nonlinear partial differential HJB equation Eq. (3.30) is extremely difficult to solve, however, when it can be solved, a candidate for an optimal control function is found as a function of state trajectory, which is a highly desirable feedback form, it also provides both the necessary and sufficient condition for the optimal cost function. While the recurrence equations of DP Eq. (3.21) provides the exact solution to a discrete approximation of the optimization equation, the HJB equation provides the approximate solution of the exact optimization equation. This HJB approach also suffers from the curse of dimensionality and the menace of the expending grid limitation like the basic approach.

3.3.3 Iterative Dynamic Programming

Iterative dynamic programming (IDP) is an extension of the traditional dynamic programming (DP) method developed by Luus in the 1990s [27]. Where in the IDP Luus used two ideas to improve the shortcoming of the traditional DP method [122], they are as follow:

Region reduction: IDP reduces the computational space and time of the traditional DP by applying it in an iterative manner while reducing grid size. In the first iteration the IDP chooses an arbitrarily small number of state and control quantization grid points by first searching over a relatively coarse but large state and control grid space range applying the traditional DP, then the resulting trajectory is used as the center to generate a denser and narrower grid space range. The successive reduction in the grid space range is then repeated in every iteration, but the number of grid points remains constant so the interval between the grid point gets narrower and the grid becomes smoother in every iteration.

Accessible states: The second idea of accessible states of IDP is to solve the menace of the expanding grid problem of the traditional DP. Instead of choosing the state grid points independently, IDP first generates the grid points as reachable states of the system by simulating the system model in Eq. (3.17) applying arbitrary quantized values of the control at each stage

[123]. Then the control domain is reduced (region reduction) from one iteration to the next, and the search process ends after a certain number of iterations.

Considering the state equation and performance index (PI) of a system as in Eqs. (3.17) and (3.18), where X_k and U_k are respectively the state and control variables. Like the traditional DP approach, the optimization process of IDP also proceeds stage-wise starting with the terminal stage $k = N$ and continues recursively down to the initial stage $k = 0$, where N is the total number of stages with $(k = 0,1,....,N)$. The basic IDP computational procedure can be described as follow [124], [125]:

1. Choose the number of grid points S.
2. Choose the number of quantized values C for each of the control variables.
3. Choose the region r_j for each of the control variables.
4. Choose S number of trail values of control inside the allowable control region to generate the X-grid at each time stage.
5. Starting at the last stage N, for each state X-grid point try all the allowable trial control values C for each of the control variables. Choose the controls U that optimize the PI for each state X-grid point and store the value of control and PI to use in step 6.
6. Step back to the stage $N-1$, then again for each X-grid point try all the allowable trial control values C for each of the control variables. Now by using the principle of optimality (PO) of the DP, the optimal control and optimal value of the PI from step 5 can be used to calculate the values of PI from the stage $N-1$ to the last stage N. Compare the values of PI for each state X-grid point and store the value of control that optimizes it along with the value of PI.
7. Repeat the procedure until stage 0. Store the control policy that optimizes the PI and also store the corresponding X-trajectory.
8. Reduce the region for allowable control values by contraction rate σ, (i.e., $r_{j+1} = \sigma r_j$), where j is the iteration index. Use the optimal control policy from step 7 as the midpoint for the allowable values for the controls at each stage.
9. Increment the iteration index j by 1 and go to step 4 and continue the procedure for a specified number of iterations.

Although the IDP solves the menace of the expanding grid problem effectively and can deal with high dimension problems, it exhibits some limitations. The IDP does not guarantee a global optimum as it only generates a certain small number of state grid points, and the termination condition is based on the number of iterations that are pre-specified by the user. By increasing

the state grid points in IDP can enhance the possibilities of obtaining the global optimum, but the curse of dimensionality returns.

3.4 Summary

In this chapter, a brief review of numerical approaches to solve the trajectory optimization problems (TOP) is presented. The numerical approaches of solving TOP are classified into three basic categories of indirect methods, direct methods, and dynamic programming. The characteristics, advantages, and disadvantages of these methods are discussed. Then the most common techniques that fall under these methods are also presented.

Chapter 4

4. Optimal Trajectory Generation from Predefined 4D Waypoint Networks

4.1 Introduction

This chapter presents the generation of optimal trajectories from predefined 4D waypoint networks by applying shortest path algorithms in graph theory. The optimal trajectories have been approximated by the path that minimizes the total link cost connecting the origin and destination in a pre-defined network. The graph methods often require large computation time and memory space but guarantee global optimal solutions. In this study, a single source shortest path algorithm was applied to generate the optimal aircraft trajectories that minimize fuel burn and aircraft total trip time between the initial and final waypoint in the networks.

This study is restricted to the climb, cruise, and descent phases of the flight and ignores the take-off and landing approach. It is also assumed that the initial and final waypoints of the trajectory are at an altitude of 3000 feet, where, in the initial waypoint the aircraft begins the climb phase and in the final waypoint the aircraft begins the landing approach. In this work, a total of three case studies with different lengths of flights (short, medium, and long-haul flights) were considered assuming zero wind conditions.

This work primarily attempts to quantify the benefits of fuel and time-optimal trajectories which were found by implying Dijkstra's shortest path algorithm. The benefit is meant to imply a reduction in fuel burn and total travel time due to using Dijkstra's shortest path algorithm to the actual unimproved flight. The key aspect of this study is a detailed comparison between actual flight trajectory and corresponding more efficient trajectory, thus giving the most realistic estimate of improvement potential.

4.2 Problem Formulation

The main goal of this study is to find the optimal path from predefined 4D waypoint networks that minimize the total link cost connecting the initial and the final waypoint. A representation of the waypoints network is shown in Fig. (4.1), where WP_1 is the initial waypoint and WP_N is the final waypoint of the network.

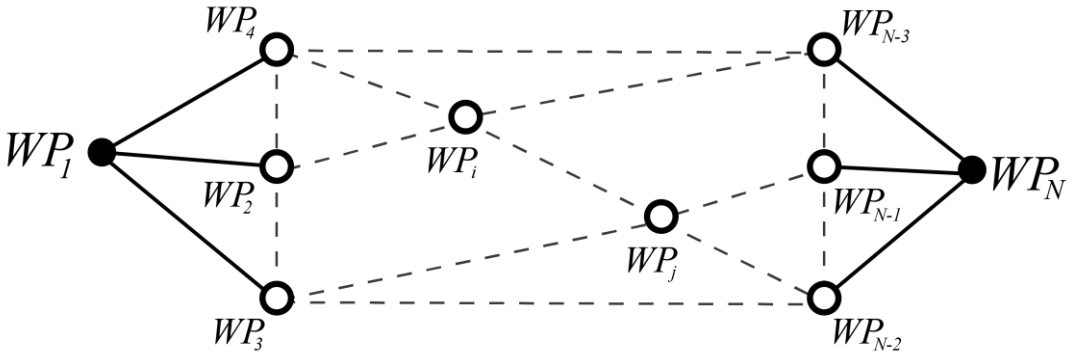


Figure 4.1: Representation of 4D waypoint networks.

Most of the approaches consider the waypoints defined by tri-dimensional coordinate positions $WP_k = (\lambda_k, \varphi_k, h_k)^T$. where, $k = 1, 2, ..., N$ and do not take into account the time. By adding the arrival time restriction to the tri-dimensional waypoint it is possible to define the 4D waypoints as $WP_k = (\lambda_k, \varphi_k, h_k, t_k)^T$. Where, $\lambda_k, \varphi_k, h_k, t_k$ are respectively longitude, latitude, altitude, and arrival time at the waypoint WP_k.

As trajectory generation typically uses a geocentric coordinates system, thus the 4D waypoints are transformed from the accustomed geodetic coordinate system to the geocentric coordinates system by the following equations [126].

$$x_k = (N_k + h_k)\cos\varphi_k \cos\lambda_k \tag{4.1}$$

$$y_k = (N_k + h_k)\cos\varphi_k \sin\lambda_k \tag{4.2}$$

$$z_k = [N_k(1-e^2) + h_k]\sin\varphi_k \tag{4.3}$$

The parameter N_k of Eqs. (4.1) – (4.3) can be calculated as follows:

$$N_k = \frac{a}{\sqrt{1-e^2\sin^2\varphi_k}} \tag{4.4}$$

In Eq. (4.4), a is the earth's semi-major axis, and e its eccentricity. Now the 4D waypoints can be demonstrated in geocentric coordinates as follows:

$$WP_k = (x_k, y_k, z_k, t_k)^T \tag{4.5}$$

The problem to be solved is to navigate the aircraft along with 4D waypoints as in Eq. (4.5) starts from the initial waypoint WP_1 to the final waypoint WP_N such that it minimizes the total fuel consumption or travel time of the aircraft.

The following section proposes a method that will determine the fuel and time-optimal path along with specified waypoints from 4D waypoint networks by implying Dijkstra's shortest path algorithm.

4.3 Proposed Method

To generate an optimal trajectory that minimizes fuel consumption or travel time between the initial and final waypoints from a set of waypoints in 4D waypoint networks requires finding the associated Fuel burn FB or travel time t by the aircraft to go from one waypoint to the other, defined as:

$$dFB_k = FF_k \times dt_k \tag{4.6}$$

Where FF_k is the fuel flow rate, dFB_k is the amount of fuel consumed, and dt_k is the amount of time needed by the aircraft to go from waypoints WP_{k-1} to WP_k and, which can be described in the following equations:

$$dFB_k = FB_k - FB_{k-1} \tag{4.7}$$

$$dt_k = t_k - t_{k-1} \tag{4.8}$$

Where, FB_k and t_k are respectively the fuel burn and travel time required to get to waypoint WP_k from the initial waypoint.

In practice, the aircraft may not pass through the waypoint WP_k exactly at the specified time t_k due to disturbances that may give rise to navigation inaccuracies. Therefore, an appropriate way is rather imposing a time tolerance interval ε at each waypoint:

$$t_k = t_{k-1} + \varepsilon \times dt_k \tag{4.9}$$

Where ε is the time tolerance interval for arrival at a determined waypoint. In the case when both waypoints WP_{k-1} and WP_k are at the same altitude, the time tolerance interval ε can be assumed as 1.

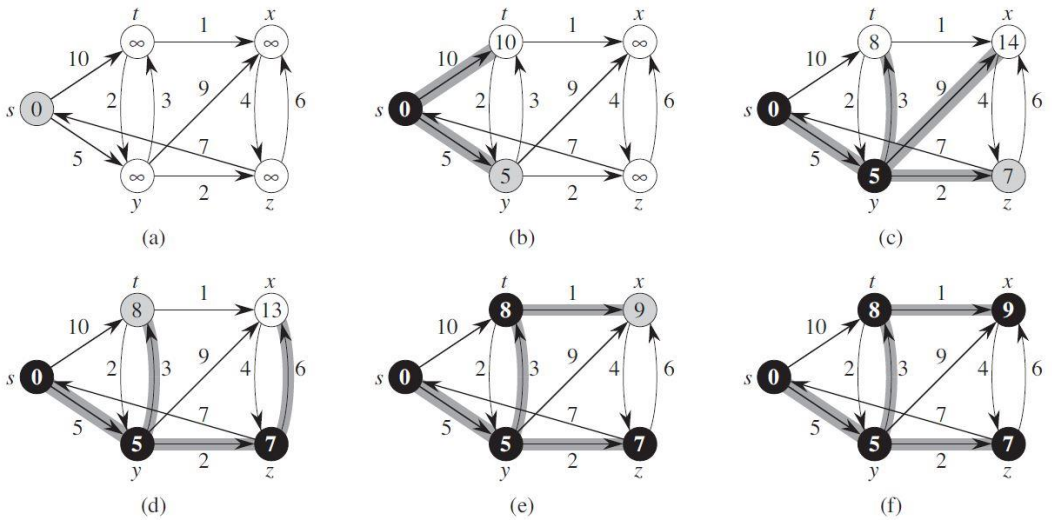
A single source shortest path algorithm called Dijkstra's Algorithm (DA) was applied to minimize the cost associated with fuel burn FB and travel time t between the initial and final waypoint. DA was first proposed by the Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959 [127], which is the most well-known shortest path algorithm. This is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest-path tree. The most common variant of the algorithm fixes one vertex as the source and another as the destination vertex and finds the shortest path between

them. DA is greatly inspired by Bellman's Principle of Optimality (PO) in the context of the shortest path algorithm [128], [129]. In this chapter, it was used as a preliminary study due to its link to Dynamic Programming (DP).

DA solves the single-source shortest-paths problem on a weighted, directed graph G (V, E) where V is a set of vertices and E is a set of edges on the graph. This algorithm requires 3 variables as input in order to find the path with the lowest cost between the source and destination vertices, they are respectively the graph, the source vertex, and the destination vertex, and at the end, it returns a reduced graph as output.

This algorithm will determine the global optimal, given a number of vertices and edges as long as it has the graph as an input, no matter how large the graph is. In addition to the basic formulation of DA, the following aspects must be defined specifically for the flight trajectory optimization problem. The number of vertices V, the edges E between the vertices, and the source and destination vertices. In this work, the waypoints of the 4D waypoint networks are the vertices V, the initial waypoint is the source vertex s, the final waypoint is the destination vertex and the associated fuel burn dFB_k and travel time dt_k by the aircraft between the pairs of waypoints are the edges E.

In Fig. (4.2) a full execution of Dijkstra's shortest path algorithm operation is shown. The circles represent the vertices or nodes and the lines with arrows are the edges. Each edge has a non-negative cost associated with it. The problem is to find the most cost-efficient route from the source vertex to any other vertex.



Source: (Cormen *et al.* pp. 659, [127])

Figure 4.2: The execution of Dijkstra's algorithm.

In this example, the source vertex s is the leftmost vertex. The value with low-cost estimates appears within the vertices, and shaded edges indicate predecessor values. Black vertices are already examined thus they have the value of the lowest cost associated with them to go from the source vertex, and the white vertices are going to be examined. The first step (a) shows the situation just before the first iteration of the while loop. From step (b) to step (f) shows the situation after each successive iteration of the while loop. The value of the lowest cost and predecessors are shown in the last step (f), and these are the final values of the lowest cost to go to that vertex from the source vertex [130], [131], [132].

4.4 Modeling of 4D Waypoints Network

Generally, the system dynamics is modeled by a set of nonlinear Equations of Motion (EOMs). In this study, a simplified version of the Three Degrees of Freedom (3DOF) EOMs are considered, where the state vector is represented by the position, velocity, flight path angle, and heading of the flight vehicle.

The following differential equations are the dynamic model used to model the problem:

$$\dot{x} = V\cos\gamma\cos\psi \tag{4.10}$$

$$\dot{y} = V\cos\gamma\sin\psi \tag{4.11}$$

$$\dot{z} = V\sin\gamma \tag{4.12}$$

$$\dot{V} = u_1 \tag{4.13}$$

$$\dot{\gamma} = u_2 \tag{4.14}$$

$$\dot{\psi} = u_3 \tag{4.15}$$

where, (x, y, z) are the geocentric coordinate system, the $V, \gamma,$ and ψ are the velocity, flight path angle, and heading respectively, the variables $u_1, u_2,$ and u_3 are respectively the acceleration, the flight path angle rate, and the heading rate. The state vector is composed by $X = [x, y, z, V, \gamma, \psi]$ and the control vector is composed by $U = [u_1, u_2, u_3]$.

The real-world flight operates under several constraints, due to aerodynamic, structural, and propulsive limitations, bound constraints are imposed on the state and control vectors as follow:

$$V^{\min} \le V \le V^{\max} \tag{4.16}$$

$$\gamma^{\min} \le \gamma \le \gamma^{\max} \tag{4.17}$$

$$\psi^{\min} \le \psi \le \psi^{\max} \tag{4.18}$$

$$u_i^{\min} \le u_i \le u_i^{\max},\ i = 1, 2, 3 \tag{4.19}$$

4.5 Simulation and Results

In this section, the simulation and results of the fuel and time-optimal trajectories are presented for three different case studies considering zero wind conditions. In the first example, a short-haul flight from Lisbon to Geneva, in the second example a medium-haul flight from Lisbon to Stockholm, and in the third example a long-haul flight from Lisbon to Montreal were considered. For all cases, the initial and final waypoints are at an altitude of 3000 feet, where, in the initial waypoint the aircraft begins the climb phase and in the final waypoint the aircraft begins the landing approach.

4.5.1 Example 1

This subsection presents the simulation and results of the first case study, where a short-haul flight from Lisbon to Geneva was considered. The 4D waypoint networks of this short-haul flight consist of two trajectories and have a total of 22 waypoints including the initial and final waypoints, and each trajectory has 12 waypoints including the initial and final waypoints. Boeing 737-700 (B737) aircraft was used to analyze the flight trajectories. The 4D waypoints in the network for the short-haul flight is presented in Fig. (4.3).

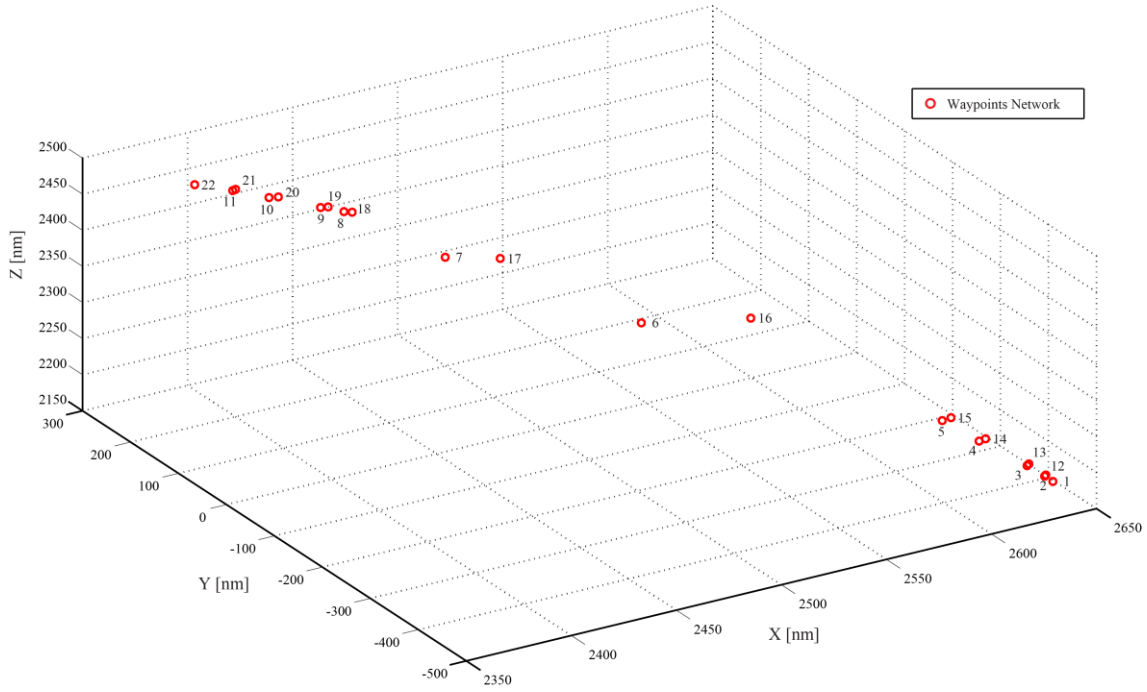


Figure 4.3: Short-haul flight 4D waypoints network.

Tables (4.1) and (4.2) Show the waypoints lists of the trajectories. Each waypoint is defined in geocentric coordinates (x, y, z) the travel time dt_k and consumed fuel dFB_k between the waypoints are also shown.

Table 4.1: List of waypoints in 1st trajectory for short-haul flight.

waypoint	x [m]	y [m]	z [m]	dt_k [min]	dFB_k [Kg]
Initial (WP_1)	2647.235288	-421.1992558	2155.785264	0	0
WP_2	2644.86006	-414.5681541	2161.780769	2.371751	291.0612
WP_3	2639.688421	-400.3464116	2173.301325	3.437191	348.0156
WP_4	2626.183424	-359.68012	2199.831101	7.295251	535.3803
WP_5	2617.384267	-321.3434642	2217.611029	5.748997	294.6361
WP_6	2513.665622	-148.0478476	2351.006175	32.48854	1176.085
WP_7	2455.327051	5.585254511	2415.971579	23.71969	858.6528
WP_8	2440.964861	153.4156637	2425.576695	19.97808	723.2065
WP_9	2432.450512	164.6407432	2431.936166	2.058735	7.823191
WP_{10}	2415.587867	198.5338065	2443.073738	5.71202	32.05871
WP_{11}	2404.139607	224.5489423	2449.734117	5.275779	42.20624
Final (WP_{22})	2389.738702	240.8587387	2460.555523	5.618537	54.83692
Total				**113.7046**	**4363.963**

Table 4.2: List of waypoints in 2nd trajectory for short-haul flight.

waypoint	x [m]	y [m]	z [m]	dt_k [min]	dFB_k [Kg]
Initial (WP_1)	2647.235288	-421.1992558	2155.785264	0	0
WP_{12}	2646.148436	-411.3007568	2160.834393	2.862798	351.3225
WP_{13}	2641.765637	-394.8066635	2171.800352	3.66612	371.1947
WP_{14}	2631.351084	-350.7310454	2195.127395	7.374258	541.1783
WP_{15}	2624.264915	-308.9768094	2211.270337	6.036433	309.3672
WP_{16}	2581.499623	-79.54841825	2280.219078	32.6647	1182.462
WP_{17}	2485.884636	24.1455534	2384.614747	23.55432	852.6664
WP_{18}	2445.165301	155.0221788	2421.268939	19.04446	689.4095
WP_{19}	2437.263718	170.4015841	2426.749665	2.415762	9.179894
WP_{20}	2421.774537	205.9872305	2436.367639	5.787862	32.48437
WP_{21}	2407.152816	231.6764483	2446.133024	5.625951	45.00761
Final (WP_{22})	2389.738702	240.8587387	2460.555523	5.642634	55.0721
Total				**114.6753**	**4439.345**

To find the fuel and time optimal trajectory from the predefined 4D waypoint networks, all possible connections between waypoints in both trajectories were established, and their travel time dt_k and consumed fuel dFB_k between these possible waypoint connections were calculated.

4.5.1.1 Fuel Optimal Trajectory

The fuel optimal trajectory was generated from the 4D waypoint networks using DA. The fuel optimal trajectory contains a total of 9 waypoints [initial waypoint (WP_1)$\rightarrow WP_2 \rightarrow WP_3 \rightarrow WP_4 \rightarrow WP_5 \rightarrow WP_{18} \rightarrow WP_{19} \rightarrow WP_{11} \rightarrow$ final waypoint (WP_{22})]. The comparison of fuel consumed in different phases of flight for these two trajectories and fuel optimal trajectory are shown in table (4.3).

Table 4.3: Fuel consumed from initial to the final waypoint in different trajectories for short-haul flight.

Trajectory	**Fuel consumed [Kg]**			**Total [Kg]**
	Climb	**Cruise**	**Descent**	
1	1469.1	2757.9	136.9	4363.9
2	1573.1	2724.5	141.7	4439.3
Fuel optimal	1469.1	2652.8	136.1	4258

As seen in the table (4.3), by using the fuel optimal trajectory for the short-haul flight (Lisbon – Geneva) the aircraft consumes 105.9 kg of less fuel than the first trajectory, which is equivalent to 2.4% less fuel than the first trajectory and consumes 181.3 kg of less fuel than the second trajectory, which is equivalent to 4.1% less fuel than the second trajectory.

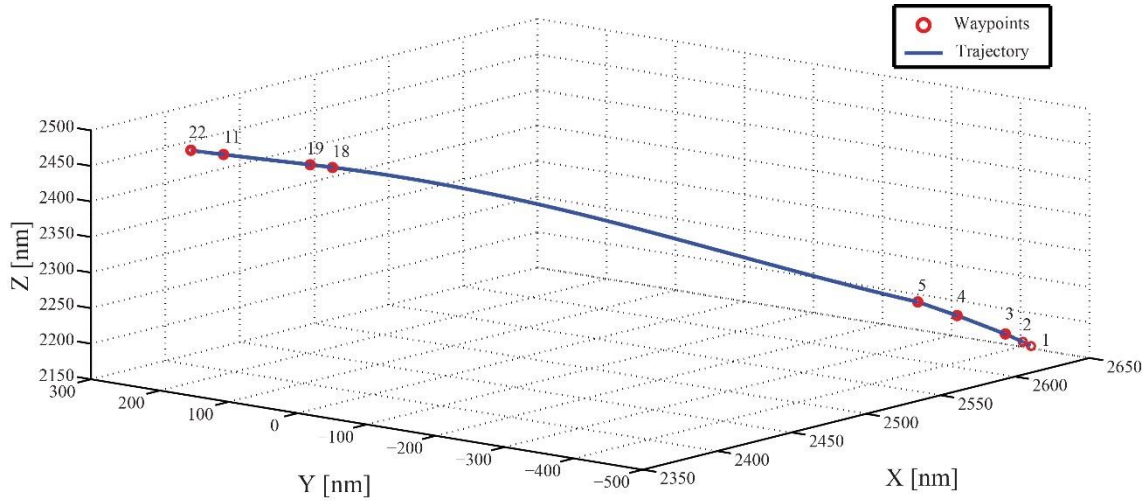


Figure 4.4: 3D fuel optimal trajectory in geocentric coordinates for short-haul flight.

The fuel optimal trajectory in 3D is shown in Fig. (4.4) where the fuel optimal trajectory is represented by the blue line, and the red circles around the trajectory are the waypoints.

4.5.1.2 Time Optimal Trajectory

The time-optimal trajectory contains a total of 10 waypoints [initial waypoint (WP_1)$\rightarrow WP_2 \rightarrow WP_3 \rightarrow WP_4 \rightarrow WP_5 \rightarrow WP_8 \rightarrow WP_9 \rightarrow WP_{10} \rightarrow WP_{11} \rightarrow$ final waypoint (WP_{22})], the distance between the initial and final waypoints in the time-optimal trajectory is 777.8 nm. The comparison of travel

time in different phases of flight for those two trajectories and time-optimal trajectory is shown in table (4.4).

Table 4.4: Time needed from initial to the final waypoint in different trajectories for short-haul flight.

Trajectory	**Time** [min]			**Total** [min]
	Climb	**Cruise**	**Descent**	
1	18.9	76.2	18.7	113.7
2	19.9	75.3	19.5	114.7
Time optimal	18.9	73.49	18.7	111.01

As it is seen from the table (4.4) that by using the time-optimal trajectory in short-haul flight (Lisbon – Geneva), the aircraft reaches the final waypoint 2.7 minutes faster than the first trajectory, which is equivalent to 2.4% of the total travel time of the first trajectory and 3.7 minutes faster than the second trajectory equivalent to 3.2% of total travel time of the second trajectory. The time-optimal trajectory of short-haul flight is shown in Fig. (4.5).

In Fig. (4.5) the blue curve line represents the real trajectory path through different waypoints, starting on the right. The red circles around the blue line denote the position of the waypoints associated with the trajectory.

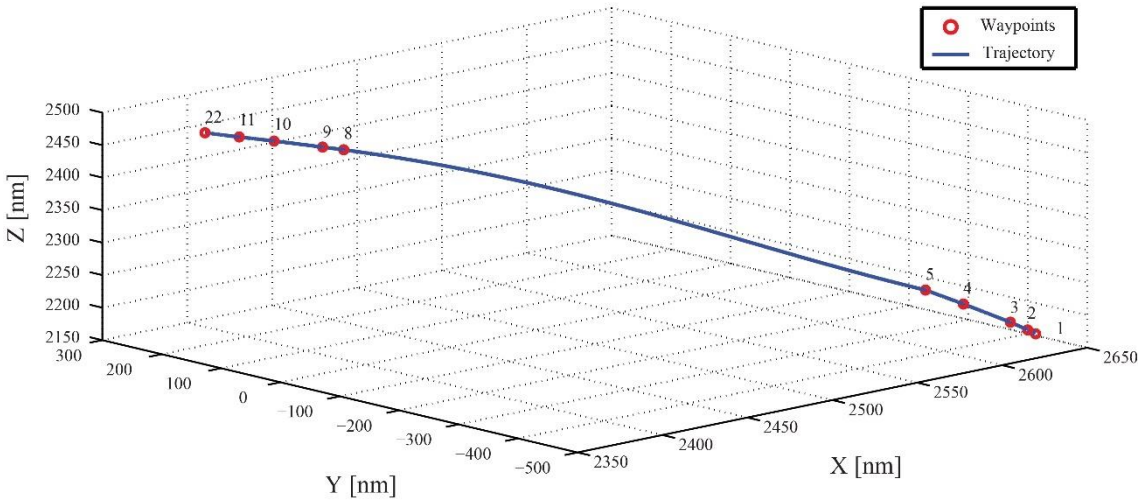


Figure 4.5: 3D time-optimal trajectory in geocentric coordinates for short-haul flight.

4.5.2 Example 2

In this example, a medium-haul flight, Lisbon to Stockholm was considered. There are also two trajectories between the initial and final waypoints in the 4D waypoint network, each trajectory has 13 waypoints including the initial and final waypoints, and a total of 24 waypoints are there in the 4D waypoint network including the initial and final waypoint. Boeing 777-200 (B772) aircraft was used to analyze the flight trajectories. The 4D waypoints in the network for the medium-haul flight is presented in Fig. (4.6).

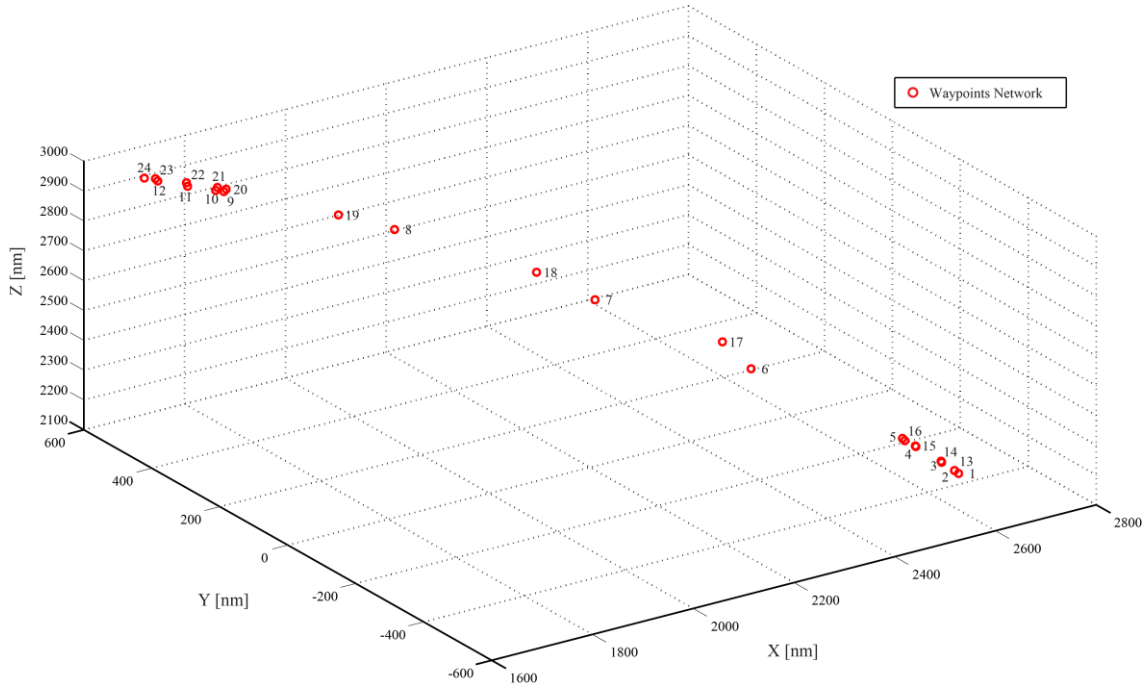


Figure 4.6: Medium-haul flight 4D waypoints network.

Tables (4.5) and (4.6) Show the waypoints lists for both of the trajectories. Each waypoint is defined in geocentric coordinates (x, y, z) the travel time dt_k and consumed fuel dFB_k between the waypoints are also shown.

Table 4.5: List of waypoints in 1st trajectory for medium-haul flight.

waypoint	x **[m]**	y **[m]**	**z [m]**	dt_k **[min]**	dFB_k **[Kg]**
Initial (WP_1)	2647.235288	-421.1992558	2155.785264	0	0
WP_2	2643.010993	-415.6490463	2163.819992	2.665515	1070.631
WP_3	2628.576005	-399.552819	2187.815353	5.529098	1724.94
WP_4	2599.967218	-365.4324126	2230.633603	8.291856	1859.863
WP_5	2586.230618	-353.564446	2249.330174	3.243938	589.0991
WP_6	2385.564946	-198.309008	2477.014631	42.43539	5385.051
WP_7	2238.11589	42.12084785	2617.270998	39.21055	4975.819
WP_8	2030.506669	324.3682322	2761.81912	47.18143	5987.324
WP_9	1782.896372	459.677681	2908.641706	39.59531	5024.645
WP_{10}	1772.18145	466.7142505	2913.251798	1.695762	31.54117
WP_{11}	1731.625729	489.6688411	2931.15743	6.807662	161.8522
WP_{12}	1690.948931	517.3407314	2947.282952	9.043191	287.5735
Final (WP_{24})	1676.867259	536.4414256	2950.540034	5.538229	212.0034
Total				**211.2379**	**27310.34**

Table 4.6: List of waypoints in 2nd trajectory for medium-haul flight.

waypoint	x [m]	y [m]	z [m]	dt_k [min]	dFB_k [Kg]
Initial (WP_1)	2647.235288	-421.1992558	2155.785264	0	0
WP_{13}	2645.498539	-411.4598298	2161.594618	2.874146	1154.43
WP_{14}	2632.238338	-391.2773722	2184.925324	5.748028	1793.241
WP_{15}	2604.511324	-357.1441614	2226.697876	8.141368	1826.109
WP_{16}	2592.536805	-337.3252479	2244.591415	3.642769	661.5269
WP_{17}	2454.226514	-11.8794812	2417.526975	49.00066	6218.184
WP_{18}	2214.11542	178.8695328	2631.773393	46.56667	5909.311
WP_{19}	1954.552046	377.266035	2809.170706	46.27695	5872.545
WP_{20}	1805.357884	485.112201	2890.720914	25.06379	3180.595
WP_{21}	1792.670589	493.4471453	2896.374754	2.016462	37.50619
WP_{22}	1747.783059	517.5842066	2916.832487	7.488843	178.0473
WP_{23}	1699.492739	535.9784473	2939.082013	9.827318	312.5087
Final (WP_{24})	1676.867259	536.4414256	2950.540034	5.864875	224.5074
Total				**212.5119**	**27368.51**

4.5.2.1 Fuel Optimal Trajectory

The fuel optimal trajectory contains a total of 9 waypoints [initial waypoint (WP_1)$\rightarrow WP_2 \rightarrow WP_3 \rightarrow WP_4 \rightarrow WP_5 \rightarrow WP_{20} \rightarrow WP_{21} \rightarrow WP_{23} \rightarrow$ final waypoint (WP_{24})], which was generated implying DA. The comparison of consumed fuel in different phases of flight for different trajectories including the fuel optimal trajectory are shown in table (4.7).

Table 4.7: Fuel consumed from initial to the final waypoint in different trajectories for medium-haul flight.

Trajectory	Fuel consumed [Kg]			Total [Kg]
	Climb	Cruise	Descent	
1	5244.5	21372.8	692.97	27310.3
2	5435.3	21180.6	752.6	27368.5
Fuel optimal	5244.5	20744.4	742.2	26731.1

From the initial waypoint to reach the final waypoint using the fuel optimal trajectory the aircraft consumes 579.2 kg of less fuel than the first trajectory and consumes 637.4 kg of less fuel than the second trajectory. In other words, by using the fuel optimal trajectory for the medium-haul flight (Lisbon – Stockholm) the aircraft consumes 2.1% less fuel than the first trajectory and 2.3% less fuel than the second trajectory. The fuel optimal trajectory in three dimensions is shown in Fig. (4.7).

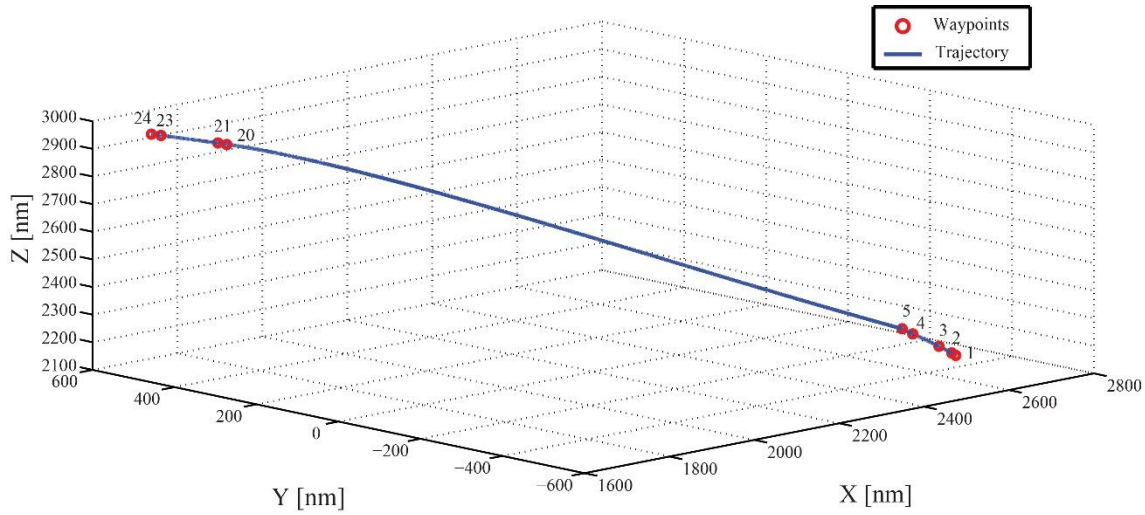


Figure 4.7: 3D fuel optimal trajectory in geocentric coordinates for medium-haul flight.

The blue curve in Fig. (4.7) corresponds to the fuel optimal trajectory for the medium-haul flight (Lisbon – Stockholm) and the red circles around the curve are the waypoints of the fuel optimal trajectory.

4.5.2.2 Time Optimal Trajectory

The time-optimal trajectory for the flight between Lisbon and Stockholm (medium-haul flight) contains a total of 9 waypoints [initial waypoint (WP_1)$\rightarrow$ $WP_2 \rightarrow WP_3 \rightarrow WP_4 \rightarrow WP_5 \rightarrow WP_9 \rightarrow WP_{10} \rightarrow WP_{12} \rightarrow$ final waypoint (WP_{24})], the distance between the initial and final waypoints in time-optimal trajectory is 1589.6 nm. The comparison of travel time in different phases of the flight for those two trajectories and the time-optimal trajectory is shown in table (4.8).

Table 4.8: Time needed from initial to the final waypoint in different trajectories for medium-haul flight.

Trajectory	**Time [min]**			**Total [min]**
	Climb	**Cruise**	**Descent**	
1	19.7	168.4	23.1	211.2
2	20.4	166.9	25.2	212.5
Time optimal	19.7	164.3	23.04	207.04

It can be seen from the table (4.8) that by flying the time-optimal trajectory in medium-haul flight (Lisbon – Stockholm) 4.2 minutes can be saved than the first trajectory, which is equivalent to 1.9% of the total travel time of the first trajectory and 5.5 minutes can be saved than the second trajectory, which is equivalent to 2.6% of total travel time of the second trajectory. The time-optimal trajectory is shown in Fig. (4.8).

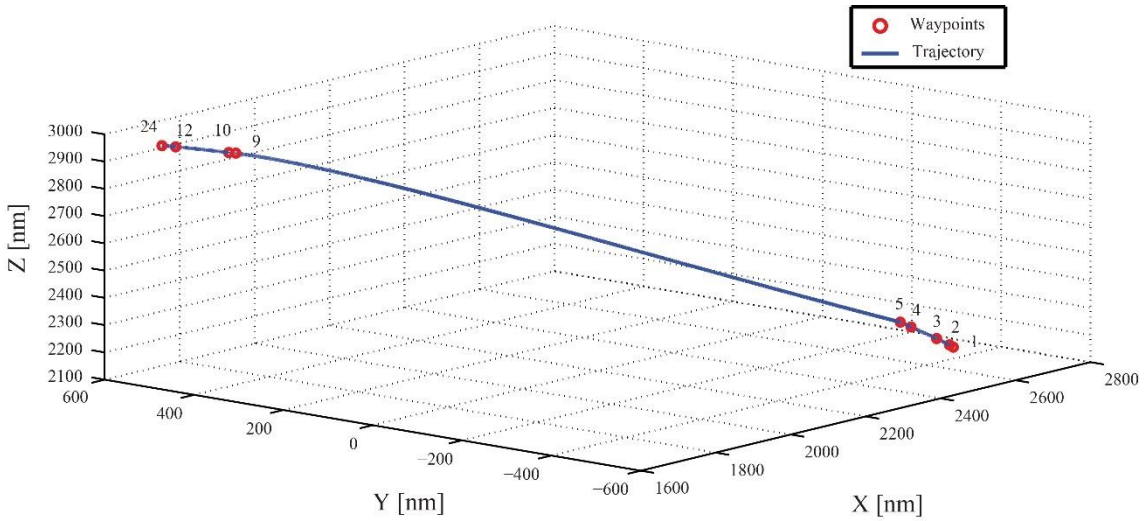


Figure 4.8: 3D time-optimal trajectory in geocentric coordinates for medium-haul flight.

In Fig. (4.8) the blue curved line is the time-optimal trajectory path and the red circles around it are the associated waypoints of the time-optimal trajectory.

4.5.3 Example 3

The flight from Lisbon to Montreal was considered in order to analyze the long-haul flight. The 4D waypoint networks of the long-haul flight also consist of two trajectories between the initial and final waypoints, each trajectory has 14 waypoints including the initial and final waypoints, and a total of 26 waypoints are in the whole 4D waypoint networks including initial and final waypoints. Boeing 777-200 (B772) was used to analyze the flight trajectories. The 4D waypoints in the network for the long-haul flight is presented in Fig. (4.9).

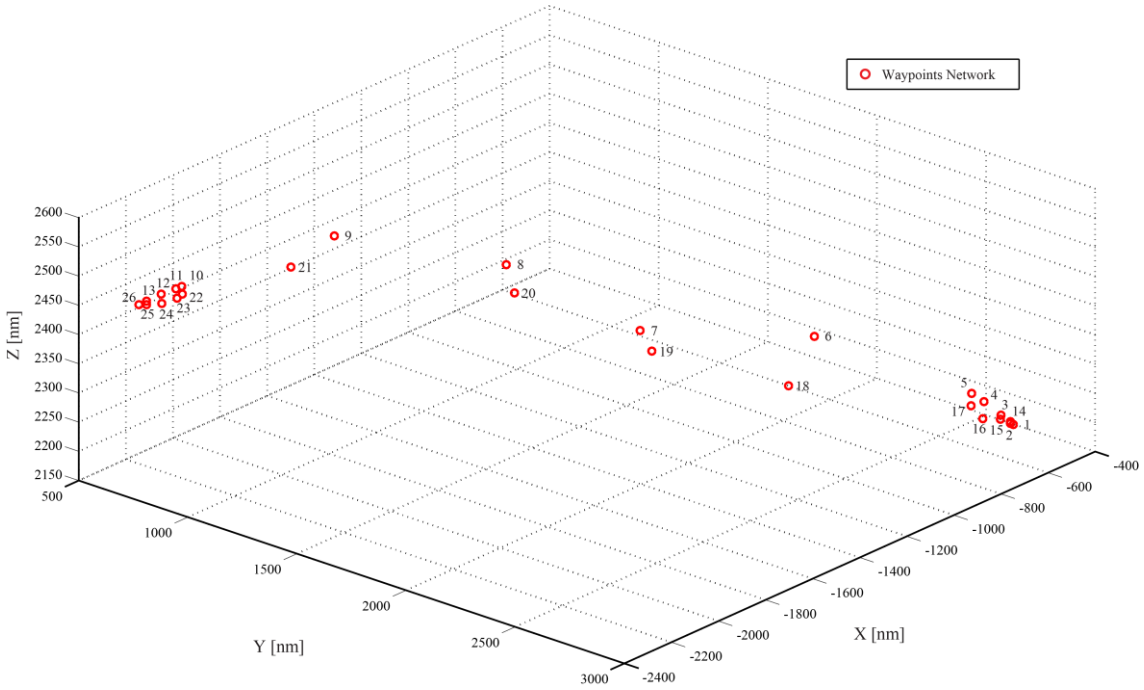


Figure 4.9: Long-haul flight 4D waypoints network.

The waypoints list of both trajectories is shown in tables (4.9) and (4.10). Each waypoint is defined in geocentric coordinates (x, y, z) the travel time dt_k and consumed fuel dFB_k between the waypoints are also shown.

Table 4.9: List of waypoints in 1st trajectory for long-haul flight.

waypoint	x [m]	y [m]	z [m]	dt_k [min]	dFB_k [Kg]
Initial (WP_1)	2647.235288	-421.199256	2155.78526	0	0
WP_1	2642.883214	-429.039923	2161.37743	2.647590246	1063.4311
WP_3	2629.148416	-456.14144	2176.10837	5.780186463	1803.27367
WP_4	2600.18989	-502.372957	2203.75409	8.209055735	1841.2912
WP_5	2579.441739	-536.054996	2221.55755	5.400149554	959.606576
WP_6	2316.256055	-959.527292	2363.48961	64.53131719	8227.74294
WP_7	2008.175978	-1413.04595	2416.741	68.56969118	8742.63563
WP_8	1654.242723	-1653.35755	2529.0831	55.05948082	7020.0838
WP_9	1189.826084	-1954.38347	2574.78207	69.12781291	8813.79615
WP_{10}	800.0516792	-2241.41972	2490.70702	61.1585403	7797.71389
WP_{11}	783.2643904	-2250.36706	2486.62851	2.421800673	43.5924121
WP_{12}	741.261309	-2272.30051	2476.63235	6.603800145	157.005348
WP_{13}	698.2019096	-2296.73195	2463.45071	8.948897858	284.574952
Final (WP_{26})	677.5684533	-2308.30204	2456.81987	5.680405215	217.445912
Total				**364.1387**	**46972.1936**

Table 4.10: List of waypoints in 2nd trajectory for long-haul flight.

waypoint	x [m]	y [m]	z [m]	dt_k [min]	dFB_k [Kg]
Initial (WP_1)	2647.235288	-421.1992558	2155.785264	0	0
WP_{14}	2644.360341	-430.1245077	2159.367752	2.514716893	1010.061187
WP_{15}	2633.016474	-461.7056605	2170.289461	6.040968112	1884.631027
WP_{16}	2616.230347	-520.8198899	2180.520245	8.362033777	1875.604176
WP_{17}	2591.647584	-549.3275236	2204.157968	5.53310668	983.2330571
WP_{18}	2326.554579	-1079.05419	2301.423495	74.72459982	9527.386477
WP_{19}	2052.376618	-1403.924042	2384.927895	53.92886276	6875.930002
WP_{20}	1703.532429	-1665.293075	2488.476927	55.77103041	7110.806377
WP_{21}	1106.19324	-2060.162823	2529.510358	89.28183768	11383.4343

WP_{22}	811.1544262	-2248.552099	2480.734122	43.99627696	5609.525312
WP_{23}	797.9082553	-2259.571568	2473.678845	2.317716218	41.71889193
WP_{24}	753.0881333	-2282.338229	2463.887157	6.983950348	166.0434195
WP_{25}	703.5924206	-2300.652959	2458.287561	9.270174258	294.7915414
Final (WP_{25})	677.5684533	-2308.302042	2456.819871	6.280806671	240.4292794
Total				**365.0061**	**47003.5951**

4.5.3.1 Fuel Optimal Trajectory

The fuel optimal trajectory of the flight between Lisbon to Montreal contains a total of 9 waypoints [initial waypoint (WP_1)$\rightarrow WP_{14} \rightarrow WP_3 \rightarrow WP_4 \rightarrow WP_5 \rightarrow WP_{22} \rightarrow WP_{11} \rightarrow WP_{13} \rightarrow$ final waypoint (WP_{26})], the distance between the initial and final waypoints in fuel optimal trajectory is 2777 nm. The comparison of fuel consumed in different phases of flight for different trajectories and the fuel optimal trajectory are shown in table (4.11).

Table 4.11: Fuel consumed from initial to the final waypoint in different trajectories for long-haul flight.

Trajectory	**Fuel consumed** [Kg]			**Total** [Kg]
	Climb	**Cruise**	**Descent**	
1	5667.6	40601.9	702.6	46972.2
2	5753.5	40507.1	742.98	47003.6
Fuel optimal	5652.1	39284.9	712.5	45649.5

Using the fuel optimal trajectory in long haul flight the aircraft consumed 1322.7 kg of less fuel than the first trajectory and consumed 1354.1 kg less fuel than the second trajectory. So, in the fuel optimal trajectory for the long-haul flight, the aircraft consumes 2.8% less fuel than the first trajectory and 2.9% less fuel than the second trajectory.

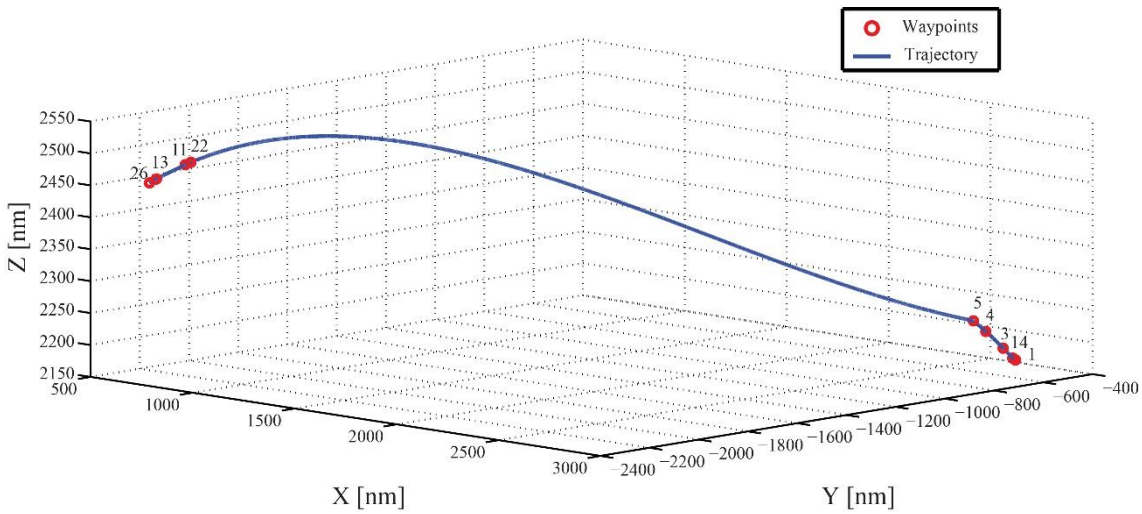


Figure 4.10: 3D fuel optimal trajectory in geocentric coordinates for long-haul flight.

The 3D fuel optimal trajectory is shown in Fig. (4.10). The blue curve in the figure is the fuel optimal trajectory and the red circles around it are the waypoints.

4.5.3.2 Time Optimal Trajectory

The time-optimal trajectory of flight between Lisbon to Montreal (long-haul flight) contains 9 waypoints [initial waypoint (WP_1)$\rightarrow WP_{14} \rightarrow WP_3 \rightarrow WP_4 \rightarrow WP_5 \rightarrow WP_{10} \rightarrow WP_{11} \rightarrow WP_{13} \rightarrow$ final waypoint (WP_{26})], the distance between the initial and final waypoints in the time-optimal trajectory of this flight is 2772 nm.

The comparison of travel time in different phases of flight for different trajectories and time-optimal trajectory is shown in the table (4.12).

Table 4.12: Time needed from initial to the final waypoint in different trajectories for long-haul flight.

Trajectory	**Time [min]**			**Total [min]**
	Climb	**Cruise**	**Descent**	
1	22.04	318.5	23.7	364.2
2	22.5	317.7	24.9	365.1
Time optimal	22.03	308.6	23.6	354.3

Table (4.12) suggests that by using the time-optimal trajectory in long-haul flight (Lisbon – Montreal) the aircraft reaches the final waypoint from the initial waypoint 9.9 minutes faster than the first trajectory which saves 2.7% of the total travel time and 10.8 minutes faster than the second trajectory which saves 2.9% of total travel time. The time-optimal trajectory of long-haul flight in 3D is shown in Fig. (4.8).

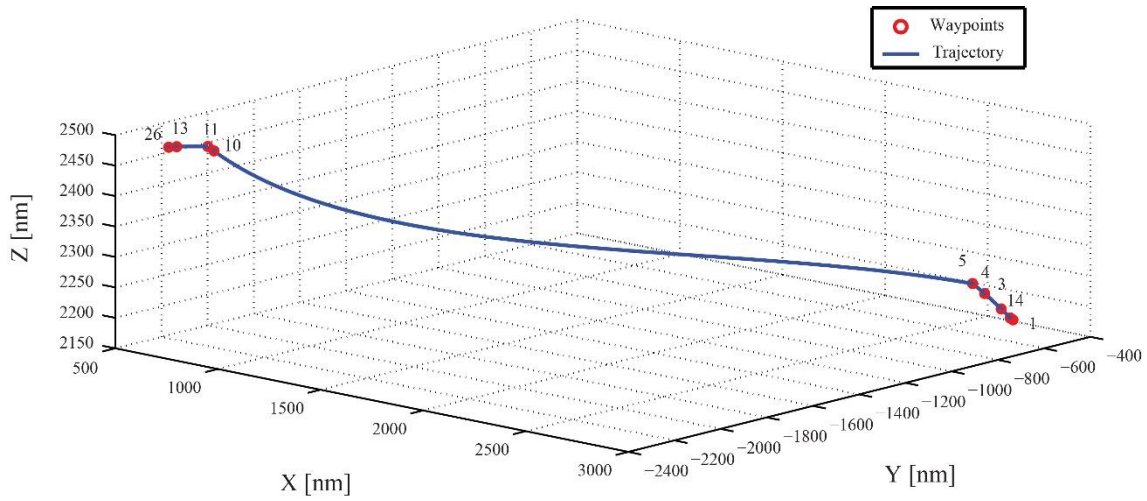


Figure 4.11: 3D time-optimal trajectory in geocentric coordinates for long-haul flight.

In Fig. (4.11) the blue curve is the time-optimal trajectory and the red circles around it are the waypoints of the trajectory. In this long-haul flight, the cruise phase is large compared to its climb

and descent phases, thus in the figure of time-optimal trajectory for this flight, the waypoints in the climb and descent phases seem too close to each other.

4.6 Summary

This study is based on finding the fuel and time-optimal trajectories of the climb, cruise, and descent phases of the flight. In this work, several steps were made in order to achieve a complete trajectory from predefined 4D waypoint networks that minimize fuel consumption and travel time. This study uses Dijkstra's shortest path algorithm as a preliminary approach to find the shortest path in a graph. The graph is referred to a collection of vertices (waypoints) and a collection of edges (associated fuel burn or travel time) that connect the pairs of vertices. This technique was used to compare different length (short, medium, and long haul) flights.

The analysis results show promising potential for reduction of consumed fuel and travel time in different flights via using Dijkstra's shortest path algorithm, across a range of common aircraft and routes. The results suggest that by flying fuel and time-optimal trajectory for a short-haul flight, it is possible to save 2.4-4.1% on fuel burn, which is equivalent to 105.9 – 181.3 kilograms of fuel and 2.7-3.7 minutes or 2.4 - 3.2 % of total travel time. In medium-haul flight by flying the fuel optimal trajectory can potentially save 2.1-2.3% fuel, reducing fuel burn by 579.2 – 637.4 kilograms, and by flying the time-optimal trajectory the travel time was reduced by 4.2-5.5 minutes or 1.9 – 2.6% of total travel time. For long haul flight, it is possible to save 2.8-2.9% on fuel burn, which is equivalent to 1322.7 – 1354.1 kilogram of fuel by flying the fuel optimal trajectory, and 9.9-10.8 minutes or 2.7 -2.9% of total travel time was saved by flying the time-optimal trajectory. In general, the savings of the fuel and time are proportional to the trip lengths and depend on the aircraft types. The results of this study are published in journals [58], [59].

Chapter 5

5. Spline Parameterization Based Nonlinear Trajectory Optimization Along 4D Waypoints

5.1 Introduction

This chapter deals with a cubic spline approximation method for solving the 4D Trajectory Optimization Problem (TOP). The state vector, its time derivative, and control vector are parameterized using Cubic Spline Interpolation (CSI). Consequently, the objective function and constraints are expressed as functions of the value of state and control at the temporal nodes, this representation transforms the TOP into a Nonlinear Programming (NLP) problem, which can be solved numerically using well-established optimization techniques.

The proposed method is successfully applied to the generation of minimum length optimal trajectories along with predefined 4D waypoints for two case studies. Where in the first case study, the take-off phase, and in the second case study the cruise phase of a typical commercial flight was considered. In both cases, zero wind condition was assumed. The key aspect of this study is a detailed comparison between nonoptimal original flight trajectory and corresponding more efficient optimal trajectory, thus giving the most realistic estimate of improvement potential.

5.2 Problem Formulation

The main goal of this study is to develop a method for solving the 4D Trajectory Optimization Problem (TOP), using cubic spline parameterization of the state and control vector based on direct optimal control approach and to validate the proposed method by generating a minimum length optimal trajectory along with predefined 4D waypoints.

Most of the approaches consider the waypoints defined by tri-dimensional coordinate positions $WP_k = (\lambda_k, \varphi_k, h_k)^T$. where, $k = 1, 2, ..., N$ and do not take into account the time. By adding the arrival time restriction to the tri-dimensional waypoint it is possible to define the 4D waypoints as $WP_k = (\lambda_k, \varphi_k, h_k, t_k)^T$. Where, $\lambda_k, \varphi_k, h_k, t_k$ are respectively longitude, latitude, altitude, and arrival time at the waypoint WP_k.

As trajectory generation typically uses a geocentric coordinates system, thus the 4D waypoints are transformed from the accustomed geodetic coordinate system to the geocentric coordinates system by using Eqs. (4.1) – (4.4) [124]. Now by using the geocentric coordinates the 4D waypoints can be defined as follows:

$$WP_k = (x_k, y_k, z_k, t_k)^T \tag{5.1}$$

The problem to be solved is to navigate the aircraft along with pre-defined 4D waypoints WP_k ($k = 1, 2, ..., N$) as in Eq. (5.1), from the initial waypoint WP_1 to the final waypoint WP_N while minimizing the total length of the trajectory.

5.3 Proposed Method

This study solves the 4D Trajectory Optimization Problem (TOP) using cubic spline parameterization. In this proposed method, the time-dependent dynamic variables (e.g., the state variables $X(t)$, its time derivative $\dot{X}(t)$, and the control variables $U(t)$) are approximate and parameterized using cubic polynomials, that transcribes the TOP into Nonlinear programming (NLP) problem, then an NLP solver is used to solve the resulting NLP and to determine the 4D optimal trajectory. The Cubic Spline Interpolation (CSI) and the parameterization of state, its derivative, and control by CSI are described in the next subsections.

5.3.1 Cubic Spline Interpolation

The Cubic Spline Interpolation (CSI) is a widely used piecewise-polynomial approximation that uses third-degree polynomials between each successive pair of nodes. The CSI is piecewise continuous which ensures that the cubic interpolant is not only continuously differentiable on the nodes but also has a continuous second derivative.

As to the mathematical spline, the points are numerical data. The weights are the interpolant coefficients on the third-degree polynomials used to interpolate the data. These coefficients shape the curve so that it smoothly passes through each of the data points [133], [134], [135].

An example of CSI is given in Fig. (5.1), with the free boundary conditions (i.e., when the second derivative of the interpolant is zero at the endpoints) the spline is also known as a natural spline. The curve of the spline of Fig. (5.1), can be visualized as if a flexible rod was forced to go through some data points. The essential idea is to fit a piecewise function of the form:

$$S_k(t) = \begin{cases} S_0(t) & if \quad t_0 \le t \le t_1 \\ S_1(t) & if \quad t_1 \le t \le t_2 \\ \vdots & \\ S_{n-1}(t) & if \quad t_{n-1} \le t \le t_n \end{cases}$$

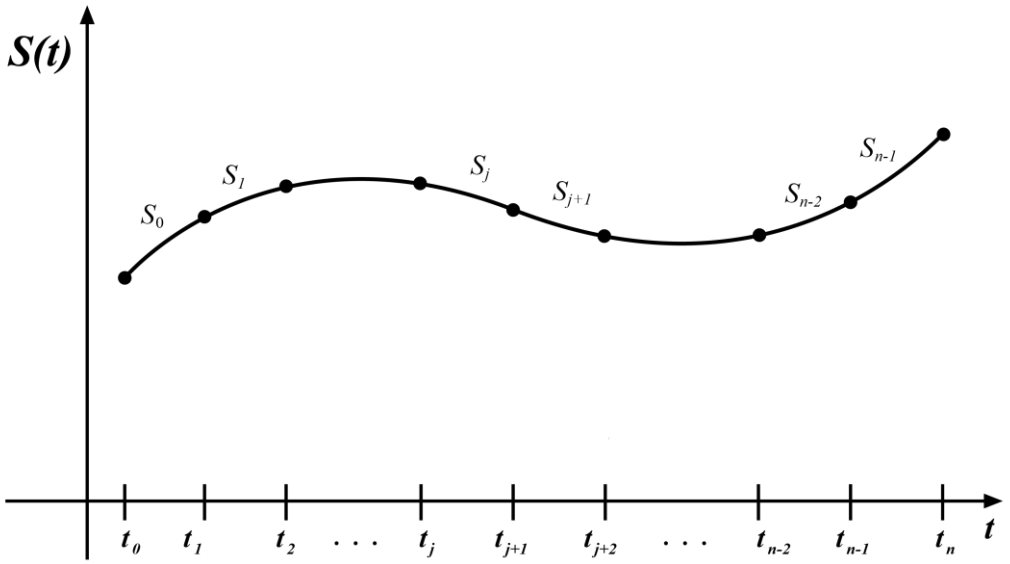


Figure 5.1: Cubic spline interpolation (CSI)

Where S_k is a third-degree polynomial of the form, for $k = 0,1,\ldots,n-1$:

$$S_k(t) = a_k + b_k(t - t_k) + c_k(t - t_k)^2 + d_k(t - t_k)^3 \tag{5.2}$$

Given a function f defined on $[a,b]$ and a set of nodes $a = t_0 < t_1 < \cdots < t_n = b$, a cubic spline interpolant S for f is a function that satisfies the following conditions:

a) $S(t)$ is a cubic polynomial, denoted $S_k(t)$, on the subinterval $[t_k, t_{k+1}]$ for each $k = 0,1,\ldots,n-1$;

b) $S_k(t_k) = f(t_k)$ and $S_k(t_{k+1}) = f(t_{k+1})$ for each $k = 0,1,\ldots,n-1$;

c) $S_{k+1}(t_{k+1}) = S_k(t_{k+1})$ for each $k = 0,1,\ldots,n-2$;

d) $S'_{k+1}(t_{k+1}) = S'_k(t_{k+1})$ for each $k = 0,1,\ldots,n-2$;

e) $S''_{k+1}(t_{k+1}) = S''_k(t_{k+1})$ for each $k = 0,1,\ldots,n-2$;

f) One of the following sets of boundary conditions is satisfied:

   i. $S''(t_0) = S''(t_n) = 0$ (natural (or free) boundary);

   ii. $S'(t_0) = f'(t_0)$ and $S'(t_n) = f'(t_n)$ (clamped boundary).

To construct a cubic spline between a pair of nodes, the 4 unknown interpolant coefficients $a_k, b_k, c_k,$ and d_k need to be determined. A spline defined on an interval that is divided into n subintervals will require determining total $4n$ interpolant coefficients.

For simplicity, all the coefficients in Eq. (5.2) can be described as Follow:

$$\begin{cases} a_k = S_k(t_k) = f(t_k) \\ b_k = \dfrac{(a_{k+1} - a_k)}{h_k} - \dfrac{h_k(2c_k + c_{k+1})}{3} \\ c_k = \dfrac{S_k''(t_k)}{2} \\ d_k = \dfrac{(c_{k+1} - c_k)}{3h_k} \end{cases} \tag{5.3}$$

Where, $h_k = (t_{k+1} - t_k)$ is the interval with $k = 0,1,\ldots,n-1$.

5.3.2 State and Control Parameterization by Cubic Splines

According to the Cubic Spline Interpolation (CSI) expression as in Eq. (5.2), the state and the control vector can be parameterized respectively on the subinterval $t \in [t_k, t_{k+1}]$ for each $k = 0,1,\ldots,n-1$.

$$SX_k(t) = a_k + b_k(t - t_k) + c_k(t - t_k)^2 + d_k(t - t_k)^3 \tag{5.4}$$

$$SU_k(t) = a_k + b_k(t - t_k) + c_k(t - t_k)^2 + d_k(t - t_k)^3 \tag{5.5}$$

Where, cubic spline interpolant $SX_k(t_k) = X(t_k)$ and $SU_k(t_k) = U(t_k)$, for the state and control vector $X(t_k) = X_k$ and $U(t_k) = U_k$. With the parameterization above, the derivative of the state vector, with respect to time can be approximated as:

$$S\dot{X}_k(t) = b_k + 2c_k(t - t_k) + 3d_k(t - t_k)^2 \tag{5.6}$$

The interpolant coefficients $a_k, b_k, c_k,$ and d_k can be determined using Eq. (5.3).

5.4 Modeling of 4D Navigation Problems

Generally, the system dynamics is modeled by a set of nonlinear Equations of Motion (EOMs). In this study, the Three Degrees of Freedom (3DOF) EOMs are considered, where the state vector is represented by the position, velocity, flight path angle, and heading of the flight vehicle. The differential equations Eqs. (4.10) – (4.15) are used to model the problem:

In these equations the three-dimensional geocentric coordinate system are defined by (x, y, z), the velocity by V, flight path angle by γ, and heading angle by ψ. The acceleration, the flight path angle rate, and the heading rate are defined by $u_1, u_2,$ and u_3 respectively. The state vector is composed by $X = [x, y, z, V, \gamma, \psi]$ and the control vector is composed by $U = [u_1, u_2, u_3]$.

The real-world flight operates under several constraints, due to aerodynamic, structural, and propulsive limitations, bound constraints are imposed on the state and control vectors as in Eqs. (4.16) – (4.19):

5.4.1 Nonlinear Programming Formulation

As discussed in the previous section the proposed method transcribes the Trajectory Optimization Problem (TOP) into a Nonlinear Programming (NLP) problem. To do this the states, its derivatives, and control vectors are parameterized by the Cubic Spline Interpolation (CSI) as shown in Eqs. (5.4) – (5.6).

Let $X = [x, y, z, V, \gamma, \psi]$ and $U = [u_1, u_2, u_3]$ be the coefficient vectors of the state and control respectively. Then expressing the trajectory optimization problem with performance index in Eq. (2.51) and such that the constraints in Eq. (2.52) – (2.54) satisfy at nodes $t_0, t_1, ..., t_f$. The performance index of the 4D TOP for minimum length trajectory can be re-stated as:

$$\min J = \int_0^{t_f} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}\, dt \tag{5.7}$$

Where the performance index J is chosen to minimize the length of the parametric curve of the 4D trajectory, and it is subjected to:

$$\begin{cases} \dot{x} - V\cos\gamma\cos\psi = 0 \\ \dot{y} - V\cos\gamma\sin\psi = 0 \\ \dot{z} - V\sin\gamma = 0 \end{cases} \tag{5.8}$$

Velocity bounded constraint:

$$V^{\min} \le V \le V^{\max} \tag{5.9}$$

Flightpath bounded constraint:

$$\gamma^{\min} \le \gamma \le \gamma^{\max} \tag{5.10}$$

Heading bounded constraint:

$$\psi^{\min} \le \psi \le \psi^{\max} \tag{5.11}$$

In this chapter the problems are considered where, the initial time t_0 and final time t_f are specified as well as the initial and final state of the problem, as shown below:

The initial conditions are:

$$[x(t_0), y(t_0), z(t_0)] = [x_0,\ y_0,\ z_0] \tag{5.12}$$

The terminal conditions are:

$$[x(t_f), y(t_f), z(t_f)] = [x_f,\ y_f,\ z_f] \tag{5.13}$$

5.5 Simulation and Results

This section presents the simulation and results of two case studies. In the first example, the take-off phase of a typical commercial flight, and in the second example, the cruise phase of a typical commercial flight was considered.

In both examples, the Trajectory Optimization Problem (TOP) was transformed into Nonlinear Programming (NLP) using the spline parameterization as described in the previous sections. Then the NLP solver *fmincon* function of the optimization toolbox of Matlab was used to solve the resulting NLP problem. All the analysis of the simulation has been done using Matlab 2016a.

5.5.1 Example 1

This subsection presents the simulation and results of the first case scenario, where the take-off phase of a commercial flight was considered. Typically, the take-off phase consists of a total of four segments, where the first, second, and final segments are climb segments and the third segment is the acceleration segment. When the aircraft reaches 35 ft above ground the first segment starts and the final segment ends when the aircraft is at a height of 1500 ft above the takeoff surface. These four segments of the flight can be represented by 5 waypoints.

Table (5.1) shows the lists of predefined 4D waypoints in a take-off phase of flight. Each waypoint is defined in the geocentric coordinates system (x, y, z) and the desired arrival time t to reach it.

Table 5.1: List of waypoints of the trajectory in take-off phase.

Waypoint	x [m]	y [m]	z [m]	t [s]
1	4914538.255	-789644.997	3974723.887	0
2	4913533.576	-788427.950	3976620.174	32
3	4912274.232	-787258.533	3979033.179	70
4	4909846.671	-780366.673	3983355.480	177
5	4908361.577	-777232.470	3987054.724	240

In this example the proposed method is successfully applied to optimize the minimum length 4D trajectory of the takeoff phase, the results are shown in table (5.2). The second column of the table (5.2) shows the distance d between pair of waypoints for the nonoptimal original trajectory which was generated by interpolation between waypoints, the third column shows the distance d_opt between a pair of waypoints for the optimized minimum length trajectory, where d_opt is the optimized objective function J (parametric curve length) as shown in Eq. (5.7), and the fourth column of the table shows the difference in distances between the original non-optimal and optimal trajectory.

Table 5.2: Distance between waypoints in take-off phase.

Waypoints	d [m]	d_opt [m]	$[d - d\_opt]$ [m]
1 - 2	2469.413	2467.081	2.332
2 - 3	2966.246	2962.444	3.802
3 - 4	8566.032	8489.586	76.446
4 - 5	5092.662	5070.812	21.850
Total	19094.353	18989.923	104.430

Table (5.2) shows the total distance from the initial to terminal waypoint for the nonoptimal original trajectory which is 19094.353 meters. It can be observed that the aircraft needs to fly 104.43 meters less if it follows the optimal minimum length trajectory which is 18989.923 meters in length. The optimal trajectory reduces the total distance by 0.55% over the nonoptimal original trajectory.

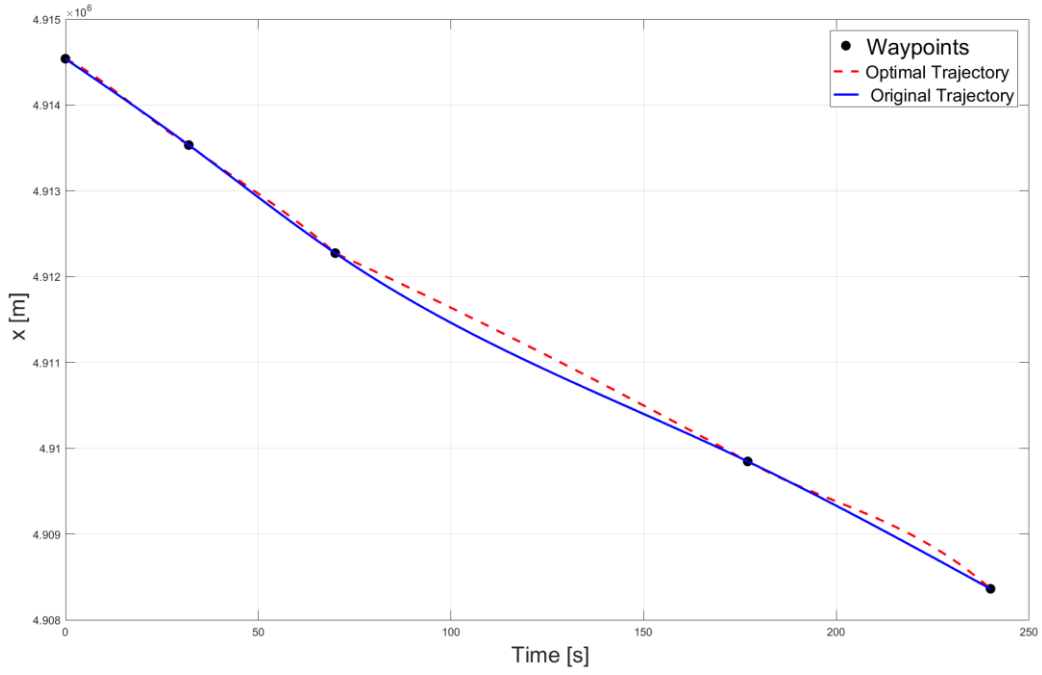


Figure 5.2: x-axis vs time in take-off phase.

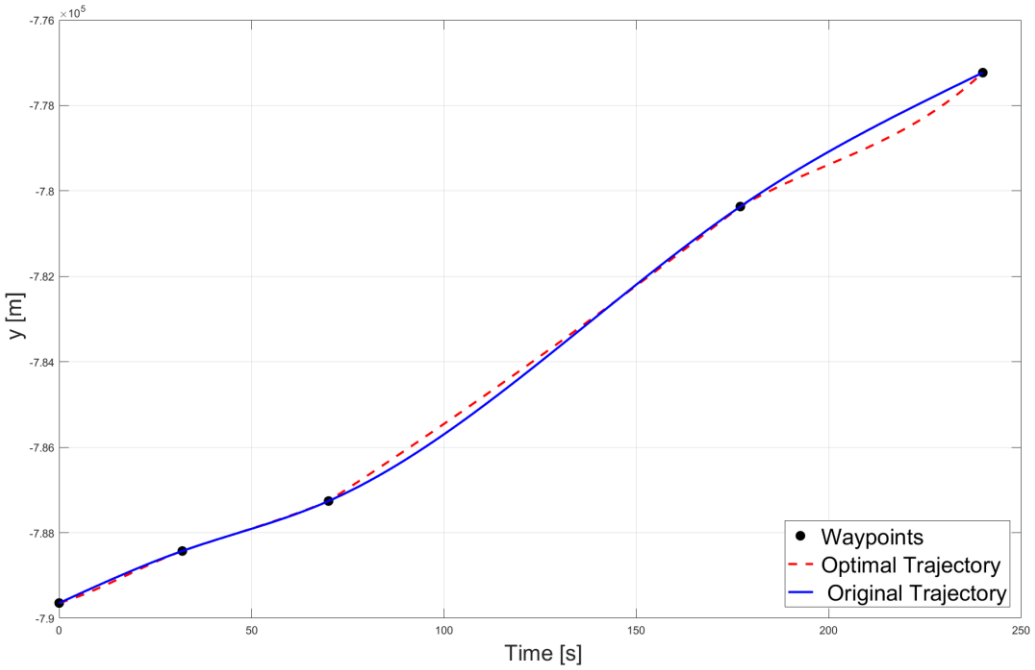


Figure 5.3: y-axis vs time in take-off phase.

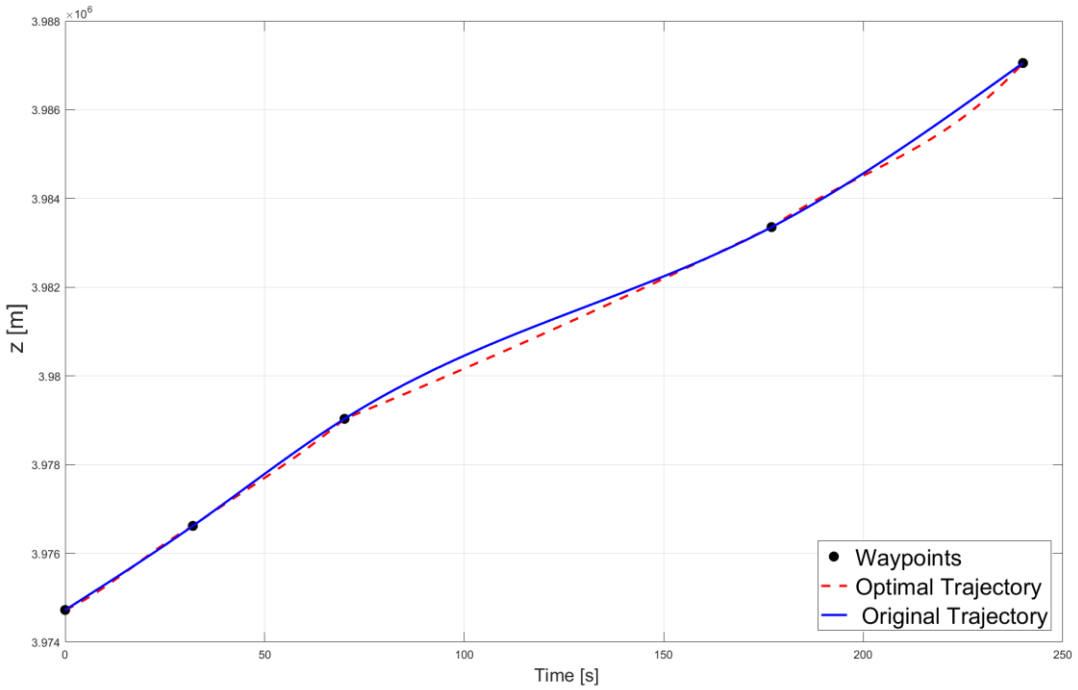


Figure 5.4: z-axis vs time in take-off phase.

Figs. (5.2) - (5-4) represent the non-optimal original trajectory as the solid blue line and the optimal trajectory as the dashed red line for the geocentric x-axis, y-axis, and z-axis vs time in meters, respectively. The solid black circles are the waypoints.

It can be seen from these figures that the generated trajectory is smooth, and the trajectory passes through all the predefined waypoints. The trajectory is also within the bounded constraints of aircraft limit as the trajectory was approximated by the cubic polynomials which not only ensures the continuity for the trajectory but also for its derivatives.

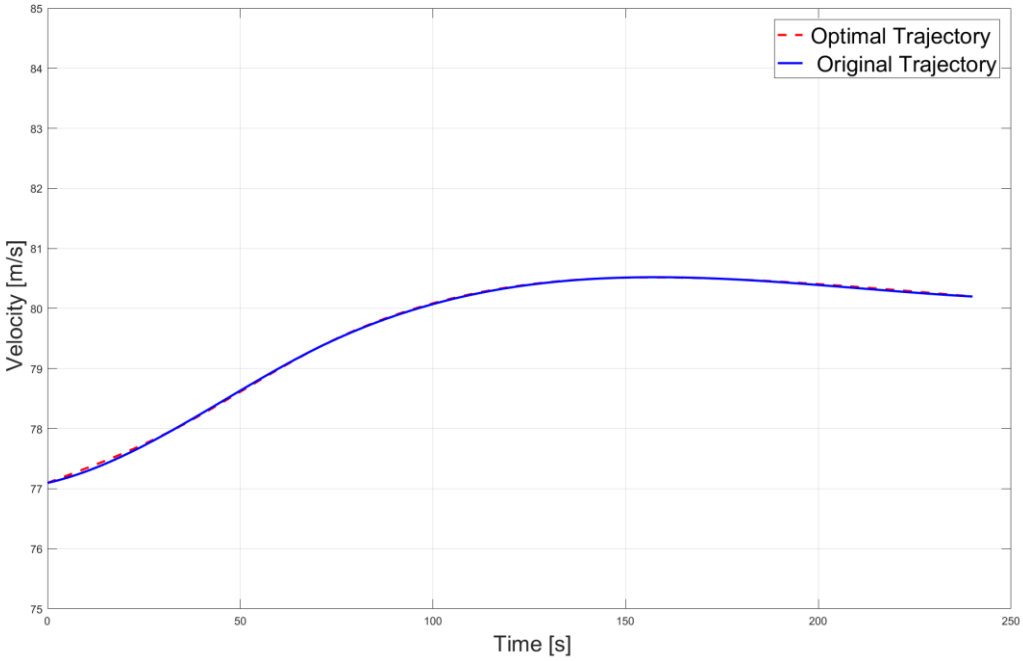


Figure 5.5: Velocity vs time of the optimal trajectory in take-off phase.

Fig. (5.5) represent the time history of the velocity in meters per second of the optimal minimum length trajectory and original nonoptimal trajectory, Fig. (5.6) represent the time history of the flight path angle in radian of the trajectories, and Fig. (5.7) represent the time history of the heading angle in radian of the optimal minimum length trajectory and original nonoptimal trajectory in climb phase.

In these figures Figs. (5.6) – (5.7), as it was predicted the velocity, flight path angle, and heading angle of the optimal trajectory satisfy the boundaries constraints that were imposed on them Eqs. (5.17) - (5.19). They also show constant behavior along the trajectory, which implies the optimal trajectory is smooth.

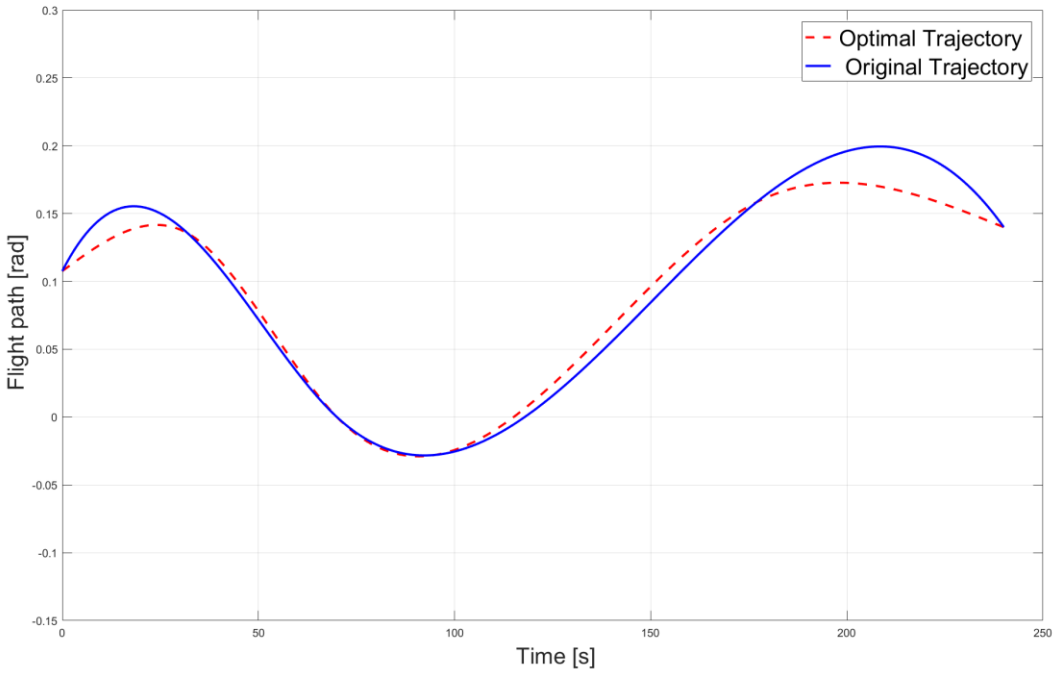


Figure 5.6: Flight path angle vs time of the optimal trajectory in take-off phase.

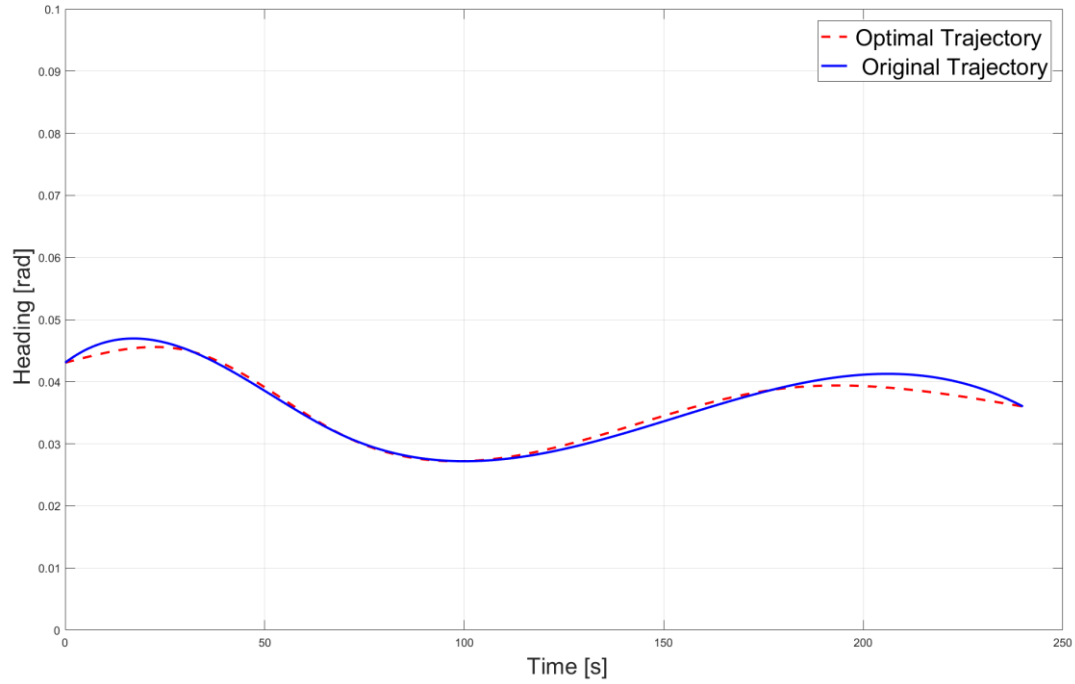


Figure 5.7: Heading angle vs time of the optimal trajectory in take-off phase.

5.5.2 Example 2

This subsection shows the simulation and results of the optimized minimum length 4D trajectory for an aircraft flying from a given initial position to a terminal position in the cruise phase, passing through some specified 4D waypoint coordinates at given scheduled times. Each waypoint is defined in the geocentric coordinates system (x, y, z) and the desired arrival time t at that waypoint is also specified.

Table (5.3) shows the lists of predefined 4D waypoints in a cruise phase of flight. Where in this study total of five 4D waypoints were chosen.

Table 5.3: List of waypoints of the trajectory in cruise phase.

Waypoint	x [m]	y [m]	z [m]	t [s]
1	4922679.656	-827307.389	3973823.239	0
2	4910474.780	-866732.905	3980948.665	180
3	4891961.555	-895195.506	3998888.720	344
4	4882512.256	-937572.651	4000554.483	528
5	4858467.696	-969054.126	4022122.186	720

In this specific mission to optimize the minimum length 4D trajectory, the proposed method achieves a solution for the problem as shown in table (5.4). Where d is the distance between pair of waypoints for the nonoptimal original trajectory, d_opt is the distance between a pair of waypoints for the optimized minimum length trajectory, and $d - d\_opt$ is the difference in distances between the original non-optimal and generated optimal trajectory.

Table 5.4: Distance between waypoints in cruise phase.

Waypoints	d [m]	d_opt [m]	$[d - d\_opt]$ [m]
1 - 2	42317.00	41882.00	435.00
2 - 3	38585.17	38404.92	180.25
3 - 4	43667.86	43450.47	217.39
4 - 5	45896.23	45104.42	791.82
Total	170466.26	168841.80	1624.50

As it is seen from table (5.4), the total distance from the initial to terminal waypoint for the original trajectory is 170466.26 meters and for the optimal minimum length trajectory is 168841.80 meters. So, the optimal trajectory reduces the total distance by 1624.50 meters from the original trajectory, which reduces the total distance by 0.95% over the original nonoptimal trajectory.

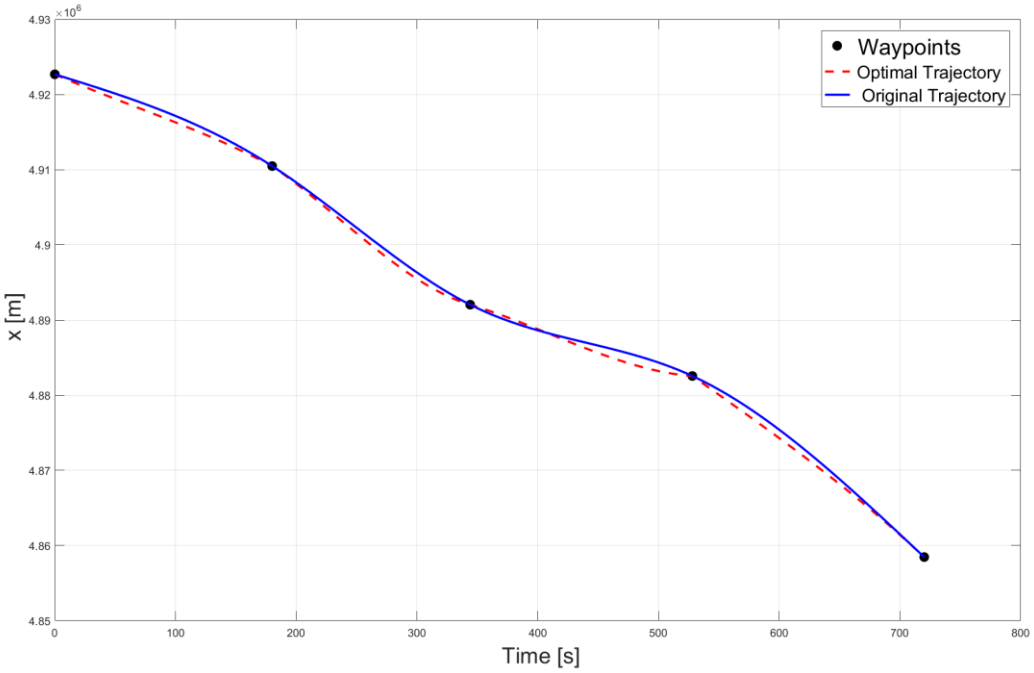


Figure 5.8: x-axis vs time in cruise phase.

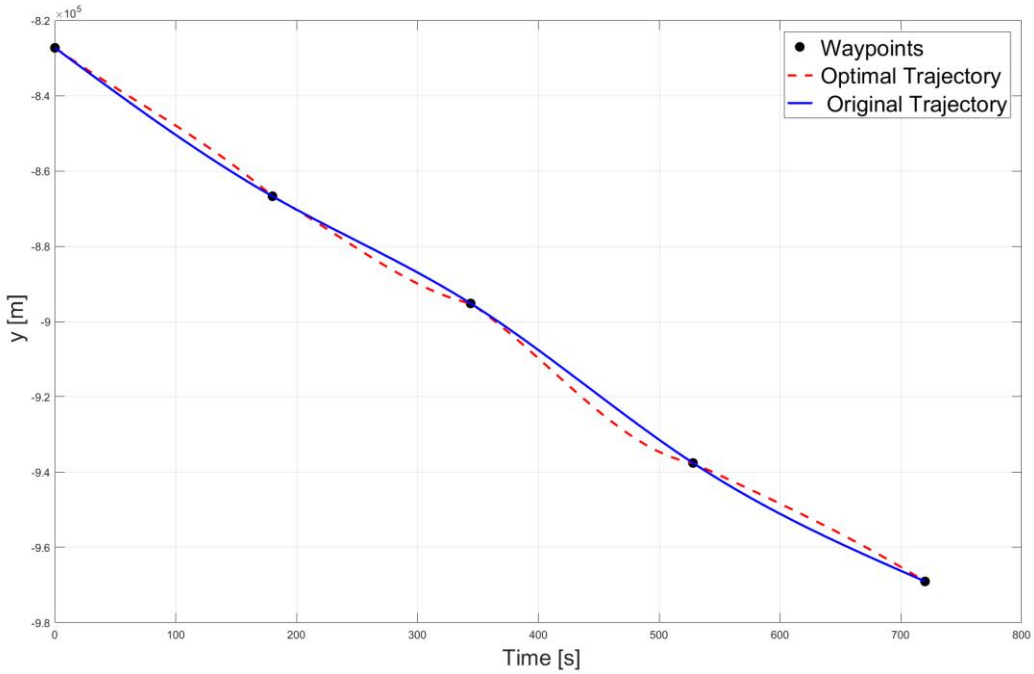


Figure 5.9: y-axis vs time in cruise phase.

Figs. (5.8) – (5.10) show the three-dimensional geocentric coordinates x-axis, y-axis, and z-axis vs time in meters, respectively of the cruise phase of flight. The optimal trajectory is represented by the dashed red line, the non-optimal original trajectory is represented by the blue line. The waypoints are represented by solid black circles. In this example of the cruise phase of flight, the

generated optimal trajectory is smooth, and the optimal trajectory passes through all the predefined waypoints.

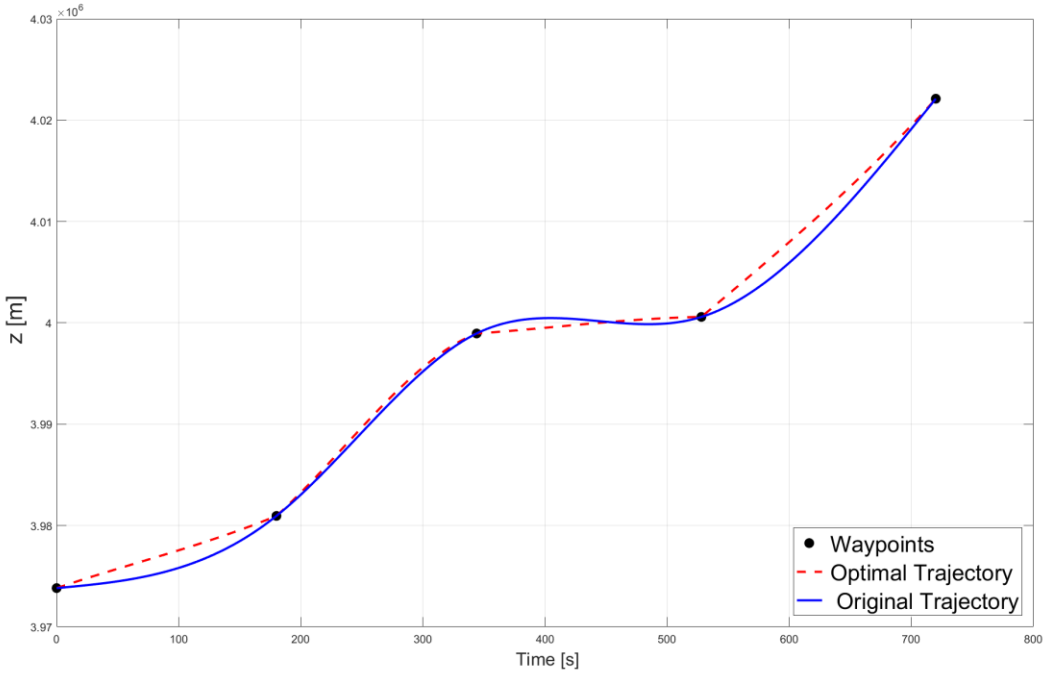


Figure 5.10: z-axis vs time in cruise phase.

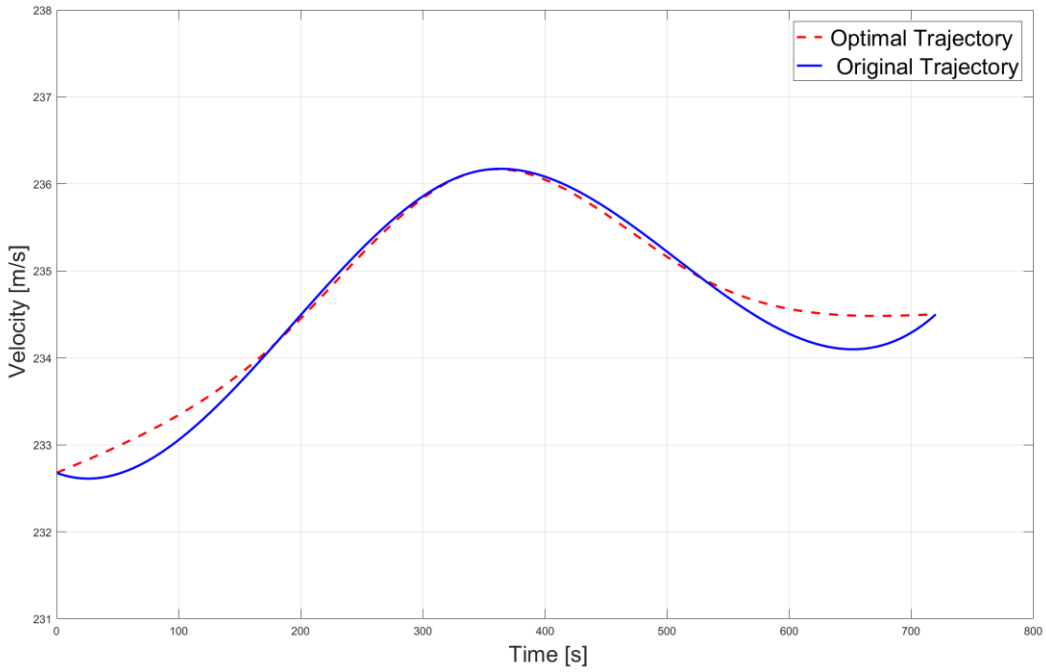


Figure 5.11: Velocity vs time of the optimal trajectory in cruise phase.

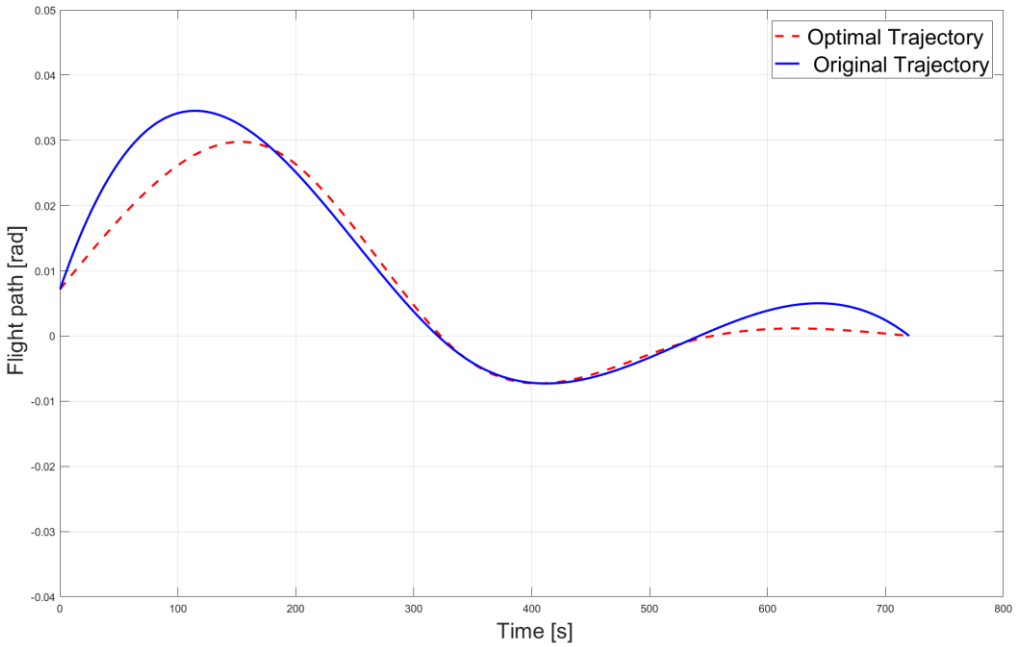


Figure 5.12: Flight path angle vs time of the optimal trajectory in cruise phase.

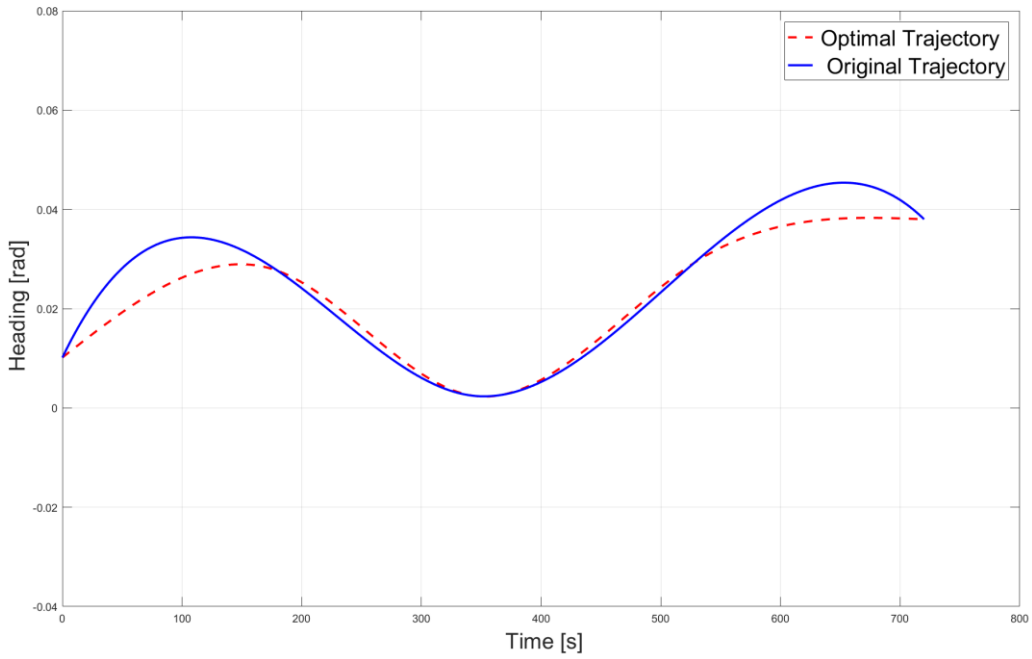


Figure 5.13: Heading angle vs time of the optimal trajectory in cruise phase.

Fig. (5.11) represent the time history of the velocity in meters per second, Fig. (5.12) represent the time history of the flight path angle in radian, and Fig. (5.13) represent the time history of the heading angle in radian of the optimal minimum length trajectory and the non-optimal original trajectory. As was predicted the figures show constant behavior along the trajectory, which

implies the generated trajectory is smooth. It can also be observed from the figures that the velocity, flight path angle, and heading angle are within the bounded constraints of the aircraft limit.

5.6 Summary

This chapter proposes a method for optimizing the 4D trajectory from pre-defined 4D waypoints. In this method, the Trajectory Optimization Problem (TOP) was solved by direct optimal control approach based on cubic spline parameterization of the state, its derivatives, and control vector, which transformed the original infinite-dimensional TOP to finite-dimensional Nonlinear Programming (NLP) problem, then the resulting NLP was solved numerically by well-established NLP solver.

In this chapter, the proposed method was applied to generate an optimal minimum length trajectory along with pre-defined 4D waypoints for the takeoff and cruise phase of flight. The proposed method generated a smooth 4D optimal trajectory with accuracy, which not only ensures the continuity for the trajectory but also for its derivatives. In the numerical examples of generating minimum length optimal trajectory from predefined 4D waypoints in the takeoff and cruise phase of flight, the method was able to reduce the total length by 0.55% and 0.95% respectively than the corresponding original nonoptimal trajectories.

The proposed method has the advantage of requiring less computer memory and computation time, which means it can be used successfully to solve online 4D trajectory optimization problems with high precision. The results of this study are published in a journal [11].

Chapter 6

6. 4D Flight Trajectory Optimization by Modified Dynamic Programming

6.1 Introduction

Dynamic Programming (DP) is a well-established numerical method to solve Trajectory Optimization Problems (TOP). The DP was first proposed by Richard Bellman in the 1950s based on a simple intuitive concept called Principle of Optimality (PO), which is used as a necessary condition of optimality to solve the TOP [18]. The numerical framework of DP is very suitable to handle discrete-time dynamic systems with nonlinear characteristics. Moreover, the 4D waypoints representation of the flight trajectory is similar to the discretization of the states grid system. Consequently, DP is a natural numerical method to deal with flight trajectory optimization. More great advantages of using DP are that it not only guarantees absolute (global) optimum but also can easily handle equality and inequality constraints of the system, and the stage-wise optimization procedure of DP is ideally suited for computers. However, the computational space and time complexity demanded by the DP is enormous due to the immense number of grid points required to find the optimum, coupled with an interpolation problem (when the trajectory from a grid point does not reach exactly the next grid point). These limitations prevent the use of the DP in many practical real-time implementations, especially with the high dimension problems.

This chapter proposes a Modified Dynamic Programming (MDP) approach to solve the flight TOP. This MDP approach reduces the computational effort and overcomes the drawbacks of the traditional DP, which allows it to be applied in high dimension problems such as 4D flight trajectory optimization problems. The preliminary results of MDP have been presented in a conference communication [136]. The computational procedure of the MDP approach is described in detail in the next section. In the later sections, the proposed method has been successfully applied to generate optimal trajectories that minimize the aircraft fuel consumption and emissions in the climb, cruise, descent phases of flight individually and also together in a global trajectory. Afterward, the obtained fuel and emissions optimal trajectories have been compared with the corresponding commercial airliner flight trajectory. The trajectory information of the commercial flight has been obtained from a flight tracking and data platform flightaware[6].

[6] Source: Flightaware . https://uk.flightaware.com//.

6.2 Modified Dynamic Programming Approach

Although traditional Dynamic Programming (DP) has many appealing features, it is not widely used in many practical applications due to the computational burden (the curse of dimensionality) and the interpolation problem (the menace of the expending grid). However, the proposed Modified Dynamic Programming (MDP) approach can be used to resolve these limitations and can be used to successfully solve 4D flight Trajectory Optimization Problems (TOP).

The computational procedure of MDP greatly reduces the computational burden of the traditional DP and resolves the limitation of the menace of the expending grid problem while retaining its appealing features. Similar to the traditional DP approach, the MDP approach is also based on the application of Bellman's Principle of Optimality (PO) concept, which allows it to split the complex optimization problem into a sequence of simple optimization subproblems and solve the problem stagewise. Mainly there are two basic differences between the traditional and modified approaches, they are the reduction of search space and the determination of the control values in each stage.

***Search space reduction*:** The MDP approach is based on the reduction of grid points at each stage, which in turn reduces the search space and required computational time. The reduction is accomplished by considering a block of grid points in each stage, instead of considering the whole state space of all possible grid points. Where the block in each stage only contains the grid points that are reachable from the grid points of the block in the previous stage. Assuming that the initial and final conditions of the problem are known, the block of the first stage only contains the grid point of the initial or final state depending on the manner (i.e., forward, or backward) of the computation procedure.

Control values generation: Another feature of the proposed MDP is that instead of applying random quantized control values at any stage the MDP approach generates the control values inside the allowable range that leads the states from a grid point to exactly a grid point at the next stage. This generation of control values eliminates the limitation of the menace of the expanding grid, as it guarantees reachable grid points for the states at any stage.

Although, because of not considering all the possible quantized states and control values in this approach, it is not possible to guarantee global optimum. However, the proposed MDP approach can successfully find the optimal trajectory within the considered region of search space and can be used to solve the real-time optimal trajectory generation problem.

6.2.1 Modified dynamic programming Algorithm

Considering a nonlinear system whose dynamics is modeled by a set of ordinary differential equations:

$$\dot{X}(t) = \mathrm{f}\left[t, X(t), U(t)\right] \tag{6.1}$$

and the performance index (PI) that need to be optimized is:

$$J = \Phi\left[t_f, X(t_f)\right] + \int_{t_0}^{t_f} L\left[t, X(t), U(t)\right] dt \tag{6.2}$$

Like the traditional Dynamic Programming (DP) the Modified Dynamic Programming (MDP) numerical procedure also requires approximating the system differential equations of a continuous system by the difference equations and approximating the integral in the Performance Index (PI) by a summation.

Thus, Eq. (6.1) can be approximated by a set of difference equations as follow:

$$X_{k+1} = \mathrm{f}_d\left[t_k, X_k, U_k\right] \tag{6.3}$$

Where, X_k and U_k are respectively the state and control vector with appropriate boundary conditions at any stage k with $(k = 0,1,...,N-1)$. The state vector X contains n state variables $X = \left[\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_n\right]$ and the control vector U contains m control variables $U = \left[\mathrm{u}_1, \mathrm{u}_2, ..., \mathrm{u}_m\right]$.

And the PI of Eq. (6.2) can be approximated by a summation as:

$$J = \Phi\left[t_N, X_N\right] + \sum_{k=0}^{N-1} L_d\left[t_k, X_k, U_k\right] \tag{6.4}$$

The equality c_{eq}, the inequality c_{inq} path constraints, and boundary constraints Ψ on the states and the controls of the trajectory are treated directly by state space and control space restriction at each decision step in the MDP approach similar to the traditional DP approach.

$$c_{eq}\left[X_k, U_k\right] = 0 \tag{6.5}$$

$$c_{inq}\left[X_k, U_k\right] \le 0 \tag{6.6}$$

$$\Psi_{\min} \le \Psi\left[X_0, X_N\right] \le \Psi_{\max} \tag{6.7}$$

This state space and control space restriction due to the path and boundary constraints are favored since it reduces the search space and required computational time in both the traditional DP and MDP approach.

Although, in most literature authors used the backward DP optimization procedure [18], [28] [72], [73], where the computations proceed stage-wise starting at the last stage $k = N$ and continues recursively down to the first stage $k = 0$. The stage-wise optimization procedure of DP can also be performed in a forward manner [22], [137], [138], starting at the first stage $k = 0$ and terminating at the last stage $k = N$. To perform DP in a backward manner, knowledge of the final state or stage number is required. However, to perform DP in a forward manner, in which the first

stage is optimized initially the knowledge of the final state or stage number is not required. Although computation time may vary between the use of the forward and backward manners, they yield the same solution [139]. Similarly, the MDP approach optimization procedure can also be performed in a forward or backward manner. In this section, the forward MDP computational procedure to solve the problem outlined in Eq. (6.3) - (6.7) is described below:

1. Define the initial condition X_0 and final condition X_N of the problem.

2. Starting at the stage $k=0$. Calculate the minimum range of each state variable $\mathrm{x}_{i,\,k+1,\min}$ and the maximum range of each state variable $\mathrm{x}_{i,\,k+1,\max}$ at the stage $k+1$, that are reachable from the initial state X_0. Where the number of state variables $(i=1,2,....,\mathrm{n})$.

3. Define the block of grid points at the stage $k+1$.

$$B_{k+1} \in \left[X_{k+1,\min}, X_{k+1,\max} \right] \tag{6.8}$$

   The states X_{k+1} inside the block B_{k+1} are considered admissible states.

4. Generate the number of grid points for each state variable $\varsigma_{i,\,k+1}$, inside the block B_{k+1} at the stage $k+1$ as follow:

$$\varsigma_{i,\,k+1} = \left[\frac{\mathrm{x}_{i,\,k+1,\max} - \mathrm{x}_{i,\,k+1,\min}}{\Delta \mathrm{x}_{i,\,k+1}} \right] + 1 \tag{6.9}$$

   Where the interval between grid points in each state variable $\Delta \mathrm{x}_{i,\,k+1}$ is selected such that the ratio $\left[\frac{\mathrm{x}_{i,\,k+1,\max} - \mathrm{x}_{i,\,k+1,\min}}{\Delta \mathrm{x}_{i,\,k+1}} \right]$ is an integer. The total number of grid points at the stage $k+1$ is $\varsigma_{k+1} = \prod_{i=1}^{n} \varsigma_{i,\,k+1}$.

5. Generate the trail control values for each control variable $\mathrm{u}_{j,\,k} \in \Omega$ that lead the state variables from a grid point at the stage k to exactly to another grid point at the stage $k+1$. Where Ω is a compact domain of feasible controls, with $(j=1,2,....,\mathrm{m})$.

6. Now at stage k, for each admissible states of X_k at this stage try all the trail control values and choose the control that optimizes the performance index (PI) Eq. (6.4), store the value of control and performance index to use in the next steps.

7. Step forward at the stage $k+1$, then again calculate the minimum range of each state variable $x_{i,\,k+2,\min}$ and maximum range of each state variable $x_{i,\,k+2,\max}$ at the stage $k+2$ that are reachable from the admissible states of X_{k+1} and repeat the procedure of step 3 to step 5. Then again for each admissible states of X_{k+1} at this stage $k+1$ try all the trail control values. Now by using Bellman's principle of optimality (PO) as the traditional DP method, the optimal control and optimal value of the performance index from step 6 can be used to calculate the values of PI from the initial stage $k=0$ to the stage $k+2$. Compare the values of PI for each state x-grid point and store the value of control that optimizes the PI, also store the value of PI.

8. Repeat the procedure until the last stage N, where the trajectory reaches the final condition X_N. This stage has only a single grid point since the final condition is specified. Store the control policy that optimizes the performance index and store the corresponding x -trajectory.

Fig. (6.1) illustrates the Stagewise optimization procedure of the MDP approach. A problem where the system has 2 state variables $X=[x_1, x_2]$ is shown in the figure. The initial state X_0 and final state X_N are shown by the red dot points at the initial stage $k=0$ and final stage $k=N$.

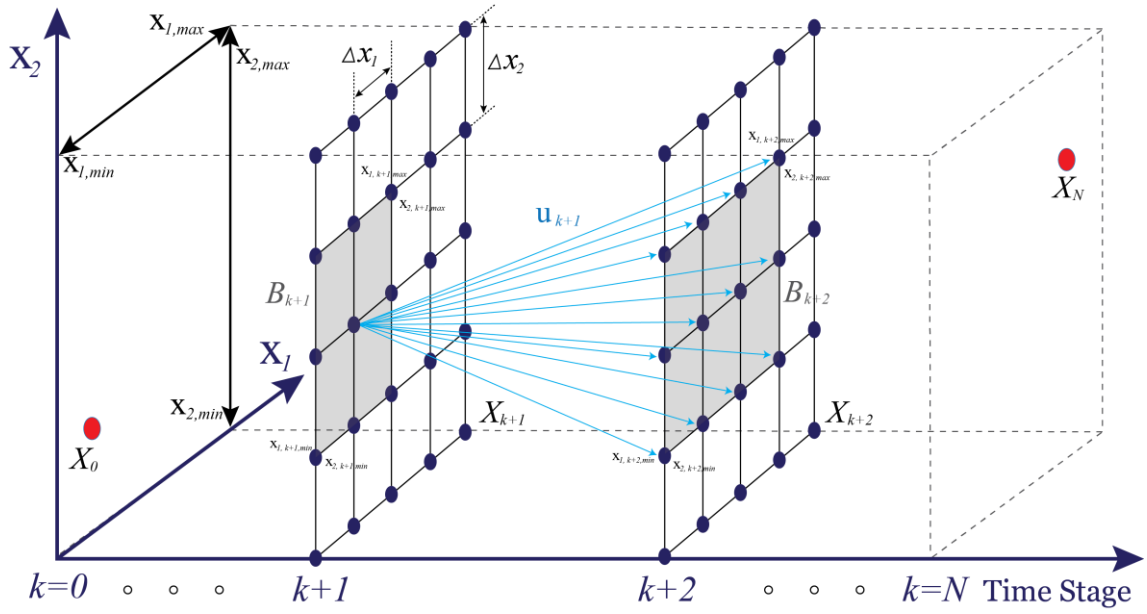


Figure 6.1: Stagewise optimization procedure of modified dynamic programming approach.

The dark blue dotted points in Fig. (6.1) are all the grid points in full state space. The highlighted grey area in each stage is the block of grid points in that stage, which is defined by the state space that is reachable from the admissible states of the previous stage. Only the grid points inside this

highlighted grey area are considered for the MDP optimization process. This reduction in search space in the optimization procedure reduces the computational space and time complexity. Fig (6.1) also illustrates the control values that are being generated in the stage $k+1$ in light blue, which guarantees that the states from a grid point in a stage $k+1$ reach another grid point in the next stage $k+2$.

This modified dynamic programming approach is effective to solve the flight trajectory optimization problem, as the flight trajectory is represented by the waypoints and the initial and final waypoint of the trajectory is generally known. The next subsection presents the computational capability of the proposed MDP approach.

6.2.2 Computational Capability of Modified Dynamic Programming

The computational procedure of the Modified Dynamic Programming (MDP) approach reduces the computational requirements of the traditional Dynamic Programming (DP), while retaining the appealing features of traditional DP.

Assuming there are n state variables and $S_{i,k}$ quantization values (i.e., grid points) for the ith state variable with $i=(1,2,...,n)$ at any stage k, The total amount of calculation of traditional DP at the stage k is given as follow:

$$S_k = \prod_{i=1}^{n} S_{i,k} \tag{6.10}$$

Where, S_k is the total number of calculations at any stage k in the traditional DP approach.

In the proposed MDP procedure, the total number of calculations is much lower than the traditional DP. If a block B_k in any stage k contains $\varsigma_{i,k}$ grid points for the ith state variable, The total amount of calculation of MDP ς_k at the stage k is given as follow:

$$\varsigma_k = \prod_{i=1}^{n} \varsigma_{i,k} \tag{6.11}$$

In the proposed MDP procedure in any stage k, the number of grid points $\varsigma_{i,k}$ for the state variables are always less than or equal to the number of grid points $S_{i,k}$ for the state variables in the traditional DP procedure as follow:

$$\varsigma_{i,k} \leq S_{i,k} \tag{6.12}$$

In most stages, the number of grid points in MDP is less than traditional DP, while the number of grid points of traditional DP is never less than the MDP. Therefore, it is guaranteed that the

total calculation of the MDP procedure ς_{Total} is less than the total calculation of the traditional DP S_{Total}:

$$\varsigma_{Total} < S_{Total} \tag{6.13}$$

Where the total number of calculations for the MDP procedure is $\varsigma_{Total} = \sum_{k=0}^{N} \varsigma_k$ and for the traditional DP procedure is $S_{Total} = \sum_{k=0}^{N} S_k$.

The proposed MDP approach can successfully find the optimal trajectory within the considered region of search space. The reduced computational space and time requirement of the MDP approach encourages its use in online optimal flight trajectory generation problems.

6.3 Trajectory Optimization by Modified Dynamic Programming

In this section, the practicality of the proposed Modified Dynamic Programming (MDP) approach is validated by generating aircraft fuel and emissions optimal trajectories between fixed initial and final 4D waypoints. In the following subsection, the aircraft Equations of Motion (EOMs) and the performance indices to minimize fuel consumption and aircraft emissions are presented. Afterward, the system differential equations are approximated by the difference equations and the integral in the performance indices are approximated by a summation in order to apply the numerical procedure of the MDP approach.

6.3.1 Modeling of 4D Navigation Problems

Generally, the aircraft system dynamics is modeled by a set of nonlinear Equations of Motion (EOMs). In this chapter, a simplified version of the Three Degrees of Freedom (3DOF) EOMs are considered, where the state vector is represented by the position, velocity, flight path angle, and heading of the flight vehicle.

The following differential equations are the dynamic model used to model the problem:

$$\dot{x} = V \cos\gamma \cos\psi \tag{6.14}$$

$$\dot{y} = V \cos\gamma \sin\psi \tag{6.15}$$

$$\dot{h} = V \sin\gamma \tag{6.16}$$

$$\dot{V} = u_1 \tag{6.17}$$

$$\dot{\gamma} = u_2 \tag{6.18}$$

$$\dot{\psi} = u_3 \tag{6.19}$$

where, (x, y, h) is the position of the aircraft, the $V, \gamma,$ and ψ are the velocity, flight path angle, and heading angle respectively, the variables $u_1, u_2,$ and u_3 are respectively the acceleration, the flight path angle rate, and the heading rate. The state vector is composed by $X = [x, y, h, V, \gamma, \psi]$ and the control vector is composed by $U = [u_1, u_2, u_3]$.

The real-world flight operates under several constraints, due to aerodynamic, structural, and propulsive limitations, bound constraints are imposed on the state and control vectors as follow:

$$V^{\min} \le V \le V^{\max} \tag{6.20}$$

$$\gamma^{\min} \le \gamma \le \gamma^{\max} \tag{6.21}$$

$$\psi^{\min} \le \psi \le \psi^{\max} \tag{6.22}$$

$$u_i^{\min} \le u_i \le u_i^{\max},\ i = 1, 2, 3 \tag{6.23}$$

Fuel consumption reduction is an important factor to improve the flight efficiency of commercial flights. In this study, the first performance index is considered to optimize the fuel consumption of the aircraft, which can be defined by the following equation.

$$J = \int_{t_0}^{t_f} \left(FF\left[h(t), V(t), \gamma(t)\right] + CI \right) dt \tag{6.24}$$

Where, $\mathrm{FF}\left[h(t), V(t), \gamma(t)\right]$ [kg/s] is fuel flow, which can be determined using Eqs. (2.27) – (2.29), t [min] is flight time, and the CI [kg/min] is Cost Index [70], it is an adjustable constant parameter that represents the cost associated with fuel burn and flight time. For all aircraft models, the minimum value (Zero) of cost index results in maximum range airspeed and minimum trip fuel but ignores the time-related cost. When the cost index is maximum, results in minimum flight time but ignores the fuel cost. In this study, the Cost index is assumed to be zero as only the fuel cost is taken into consideration.

Principle aircraft emissions (AE) that impact the environment most are carbon dioxide $\left(\mathrm{CO_2}\right)$, water vapor $\left(\mathrm{H_2O}\right)$, sulfur dioxide $\left(\mathrm{SO_2}\right)$, oxides of nitrogen $\left(\mathrm{NO_x}\right)$, carbon monoxide $\left(\mathrm{CO}\right)$, and hydrocarbons $\left(\mathrm{HC}\right)$. In this study, the sum of all the aircraft emissions is minimized in order to reduce the environmental impact caused by the aircraft. So, the AE can be described as follow:

$$AE = E_{CO_2} + E_{H_2O} + E_{SO_2} + E_{NO_x} + E_{CO} + E_{HC} \tag{6.25}$$

The second performance index that needs to be optimized to reduce aircraft emissions can be defined by the following equation.

$$J_{AE} = \int_{t_0}^{t_f} AE(t)\ dt \tag{6.26}$$

Typically, these aircraft emissions AE are modeled by the Emission Index (EI), which has units of grams of emission per kilogram of fuel burned. So, Eq. (6.26) can be rewritten as follow:

$$J_{AE} = \int_{t_0}^{t_f} EI_{AE}(t)\ FF\left[h(t), V(t), \gamma(t)\right]\ dt \tag{6.27}$$

The Emission Index (EI) of the CO_2, H_2O and SO_2 are given in table (2.1) and the EI of NO_x, CO and HC are given in Eqs. (2.37) – (2. 39).

Similar to the basic DP approach, in order to apply the numerical procedure of the MDP approach consists of approximating the system differential equations by the difference equations and approximating the integral in the performance index by a summation. The following difference equations can be used to model the system dynamics :

$$x(k+1) = x(k) + \Delta t V(k) \cos\gamma(k) \cos\psi(k) \tag{6.28}$$

$$y(k+1) = y(k) + \Delta t V(k) \cos\gamma(k) \sin\psi(k) \tag{6.29}$$

$$h(k+1) = h(k) + \Delta t V(k) \sin\gamma(k) \tag{6.30}$$

$$V(k+1) = V(k) + \Delta t u_1(k) \tag{6.31}$$

$$\gamma(k+1) = \gamma(k) + \Delta t u_2(k) \tag{6.32}$$

$$\psi(k+1) = \psi(k) + \Delta t u_3(k) \tag{6.33}$$

And the performance indices:

$$J = \Delta t \sum_{k=0}^{N-1} FF[h(k), V(k), \gamma(k)] \tag{6.34}$$

$$J = \Delta t \sum_{k=0}^{N-1} AE(k) \tag{6.35}$$

Where Δt is the incremental time. All boundary and algebraic path constraints Eqs. (6.20) - (6.23) of the system are treated directly by state space and control space restrictions.

The MDP approach also solves the TOP problem by using Bellman's principle of optimality (PO) as a necessary condition. In this study, the MDP is performed in a forward manner since the performance indexes Eq. (6.34) and (6.35) are dependent on the aircraft weight, and the final aircraft weight of the trajectory is not known a priori.

In a forward manner, assuming that the optimal controls, states, and costs are known from the initial stage 0 to any stage k. Then at any stage $k+1$ applying the principle of optimality, the performance indexes Eq. (6.34) and (6.35) can be rewritten as:

$$J_{k+1}^{*} = \min_{u(k+1)} \left[\Delta t \; FF[h(k+1), V(k+1), \gamma(k+1)] + J_{k}^{*} \right] \tag{6.36}$$

$$J_{k+1}^{*} = \min_{u(k+1)} \left[\Delta t \; AE(k+1) + J_{k}^{*} \right] \tag{6.37}$$

Where J_{k}^{*} represents the optimal cost from the initial stage 0 to any stage k, and J_{k+1}^{*} represents the optimal cost from the initial stage 0 to any stage $k+1$.

6.4 Simulation and Results

This section presents the simulation and results of the aircraft fuel and emissions optimal trajectories generated by the Modified Dynamic Programming (MDP) approach for two case studies, each case study contains four examples. The first case study is based on obtaining the fuel optimal trajectories, it consists of four examples they are respectively the climb phase, cruise phase, descent phase, and global trajectory of a commercial flight from Lisbon to Paris.

The second case study is based on obtaining the minimum aircraft emissions trajectories, this case study also consists of four examples they are respectively the climb phase, cruise phase, descent phase, and global trajectory of a commercial flight from Lisbon to Munich.

In all examples, the optimal trajectories were compared with a reference commercial flight trajectory for the same route. The flight information was taken from the website Flightaware. This website allows tracking a flight online, and the flight data are available for free. For each flight, the aircraft type, time, position, orientation, speed, and altitude are provided. However, the website does not provide the take-off weight of the aircraft.

The fuel consumption model was exported from the Base of Aircraft Data (BADA) model version 3.4 developed by [64], [140]. The BADA model consists of parametric models such as the atmosphere model, aerodynamic model, engine thrust model, and fuel consumption model which are necessary for aircraft performance calculations. It provides concrete parameters for almost all types of aircraft. Since the same BADA model is used to calculate the fuel consumption of both commercial flight trajectory and proposed optimal trajectory, the model error does not directly affect the difference in fuel consumption.

The aircraft emissions model was exported from the Boeing Fuel Flow Method 2 (BFFM2). The BFFM2 uses the International Civil Aviation Organization (ICAO) emission databank to determine the reference emission indices, which eventually allow the calculation of the aircraft emissions [67]. Since the same model is used to calculate the aircraft emissions of both commercial flight trajectory and proposed optimal trajectory, the model error does not directly

affect the difference of emissions. All the analysis of the simulation has been done using Python 3.7.

6.4.1 Fuel Optimal Trajectory

In this first case study, a flight from Lisbon to Paris and a twinjet aircraft was considered to analyze the fuel optimal trajectories. In this study, the reference mass of this aircraft was considered as the take-off weight, which is 60,000 [kg] and it is assumed there is no wind condition. The performance operational data of the aircraft is provided in Annex A.1. In all four examples, to determine the potential benefit of reduction of fuel consumption, the obtained fuel optimal trajectories were compared with a commercial flight trajectory for the same route. The constraints of the case study have been selected according to the reference commercial flight trajectory for the same route as follow:

Altitude [m]	:	$1000 \leq h(t) \leq 12500$
True airspeed [m/s]	:	$120 \leq V(t) \leq 253$
Flight Path Angle [rad]	:	$-0.1 \leq \gamma(t) \leq 0.13$
Heading Angle [rad]	:	$0.4 \leq \psi(\mathrm{t}) \leq 1.5$

In all four examples of this case study the interval between the states grid points (x, y, h) are $\Delta x = 1000$ [m], $\Delta y = 1000$ [m], $\Delta h = 50$ [m], and the interval between the stages is $\Delta t = 60$ [s].

6.4.1.1 Climb Phase

The first example of the case study is a climb phase of flight. The initial and final waypoints of the problem are set identical to the initial and final waypoints of the climb phase of the reference commercial flight trajectory. The initial and final 4D waypoints (x, y, h, t) of the problem are shown in table (6.1).

Table 6.1: Initial and final waypoints of the trajectory in climb phase.

Waypoint	***x* [m]**	***y* [m]**	***h* [m]**	***t* [s]**
Initial	4912063.702	-784235.333	1050	0
Final	4842143.116	-614830.488	11500	1080

Figs. (6.2) – (6.9) present the comparison between fuel optimal vs reference trajectory in the climb phase of flight where the orange line represents the optimal trajectory, and the blue line represents the reference commercial trajectory as exported from the website Flightaware. Fig. (6.2) shows the three-dimensional (x, y, h) position of the reference and optimal trajectory in the climb phase of flight, where the Top of Climb (ToC) is achieved earlier in the optimal trajectory than the reference trajectory in order to save fuel.

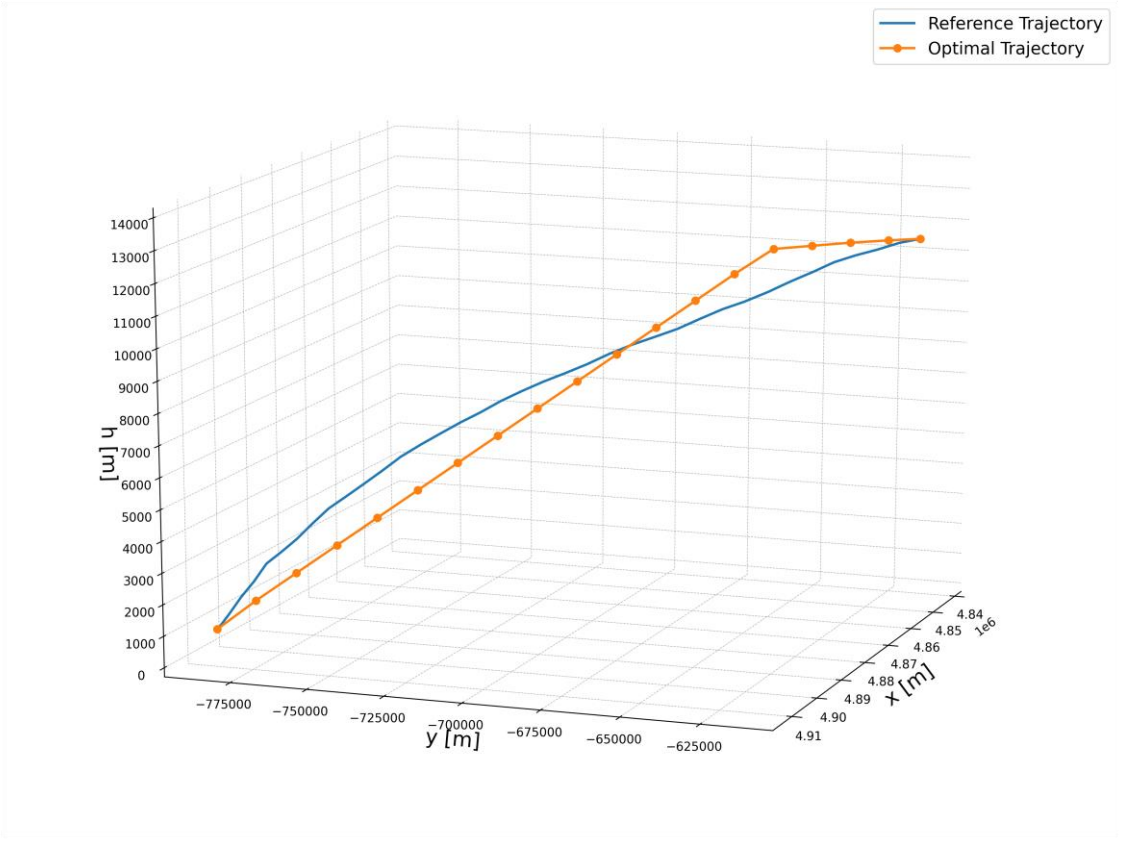


Figure 6.2: Comparison of reference and optimal 3D trajectory in climb phase.

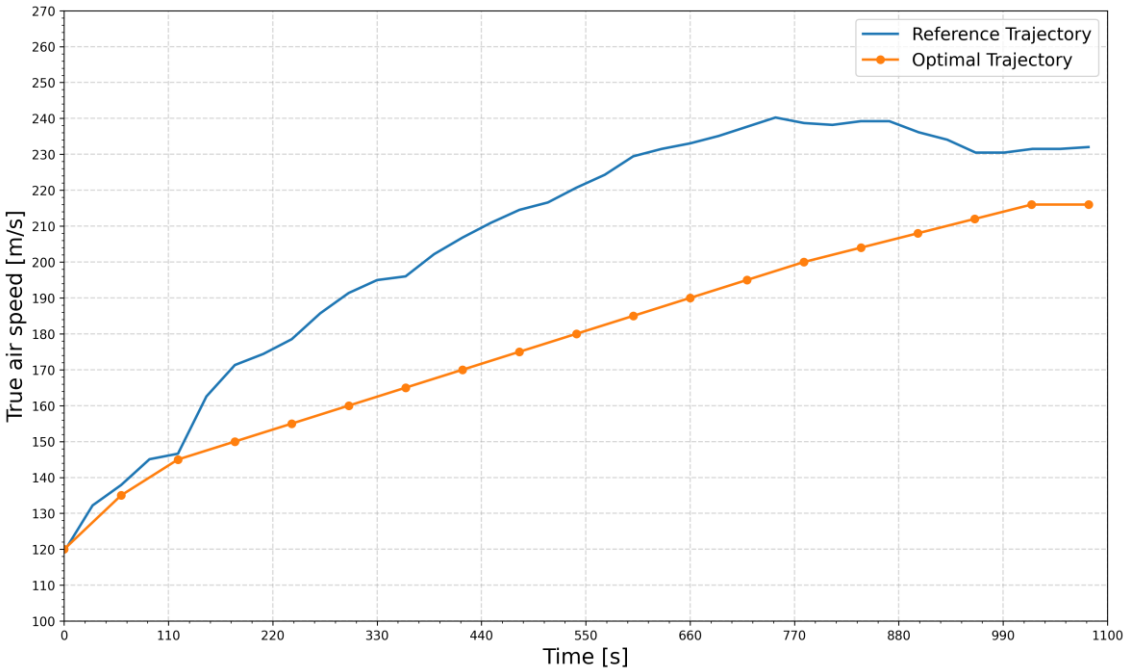


Figure 6.3: Comparison of true airspeed vs time of reference and optimal trajectory in climb phase.

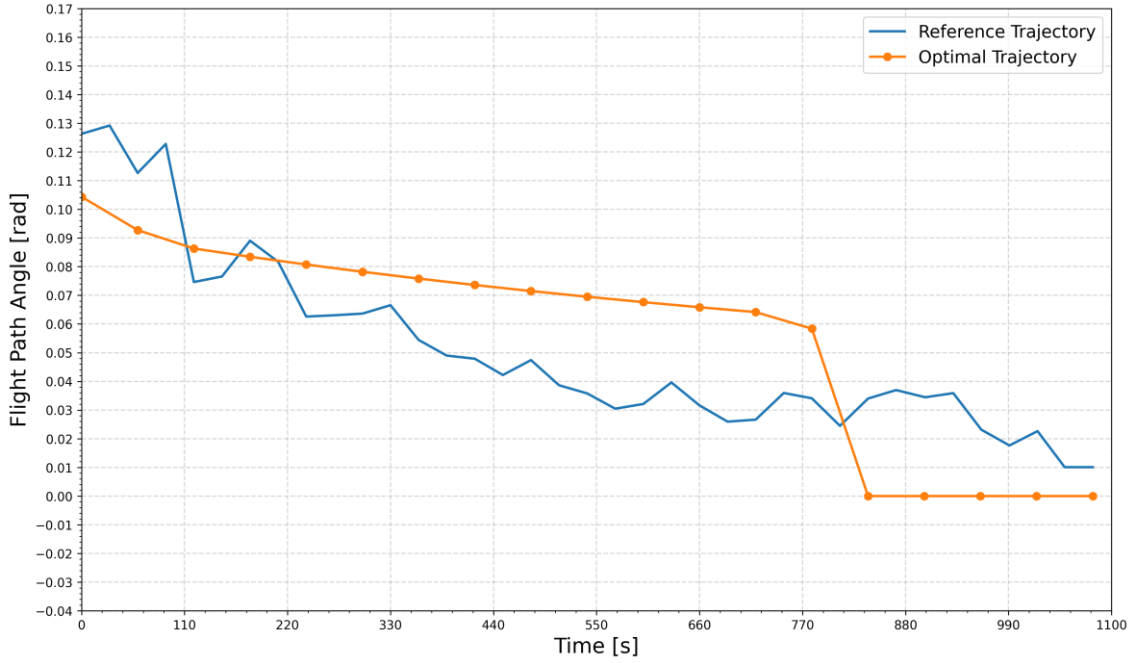


Figure 6.4: Comparison of flight path angle vs time of reference and optimal trajectory in climb phase.

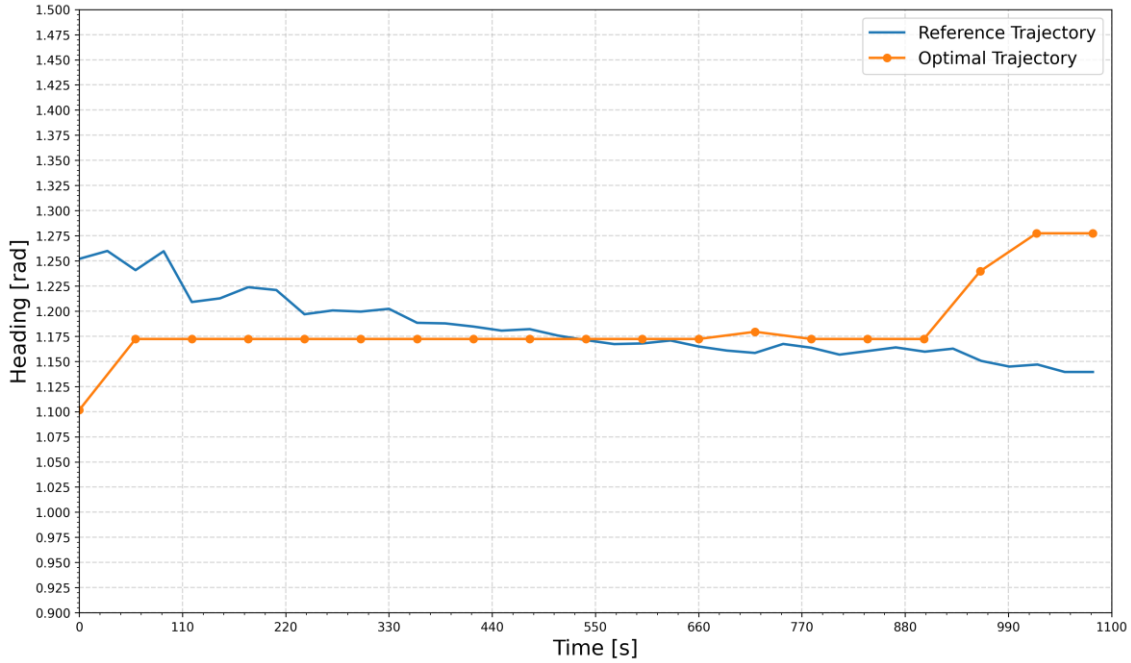


Figure 6.5: Comparison of Heading vs time of reference and optimal trajectory in climb phase.

Fig. (6.3) shows the comparison of true airspeed in $[\mathrm{m/s}]$, Fig. (6.4) shows the time history of the flight path angle in $[\mathrm{rad}]$ and Fig. (6.5) shows the heading angle in $[\mathrm{rad}]$ between the reference

and fuel optimal trajectory. In these figures, the true airspeed, flight path angle, and heading angle of the optimal trajectory are within the bounded constraints that were imposed on them.

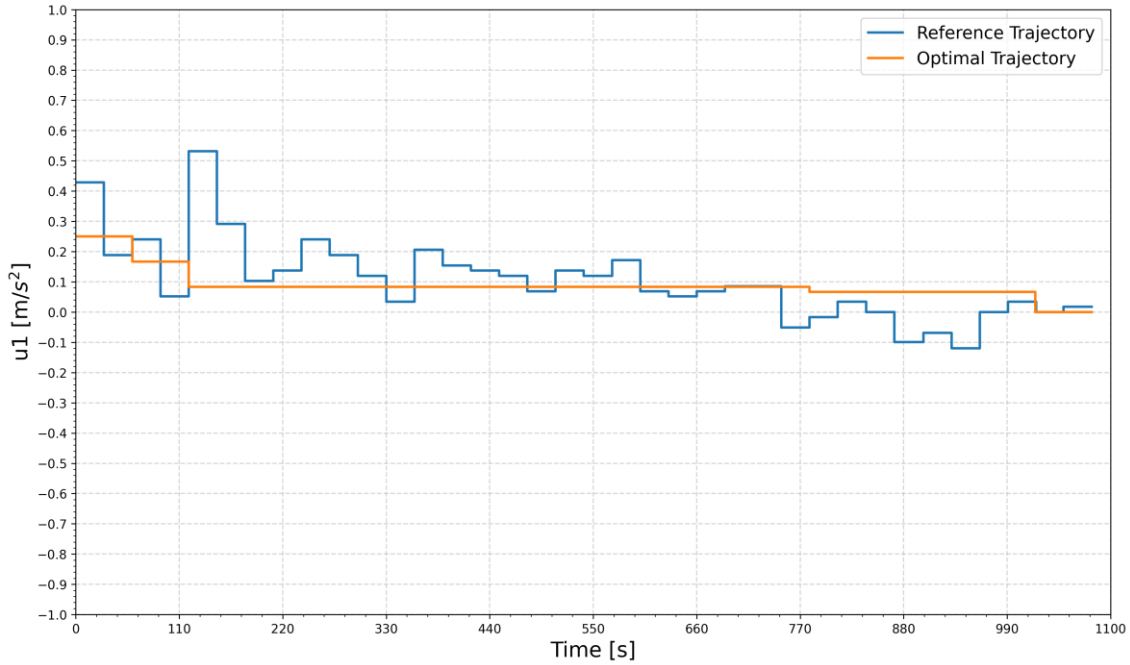


Figure 6.6: Comparison of control u1 vs time of reference and optimal trajectory in climb phase.

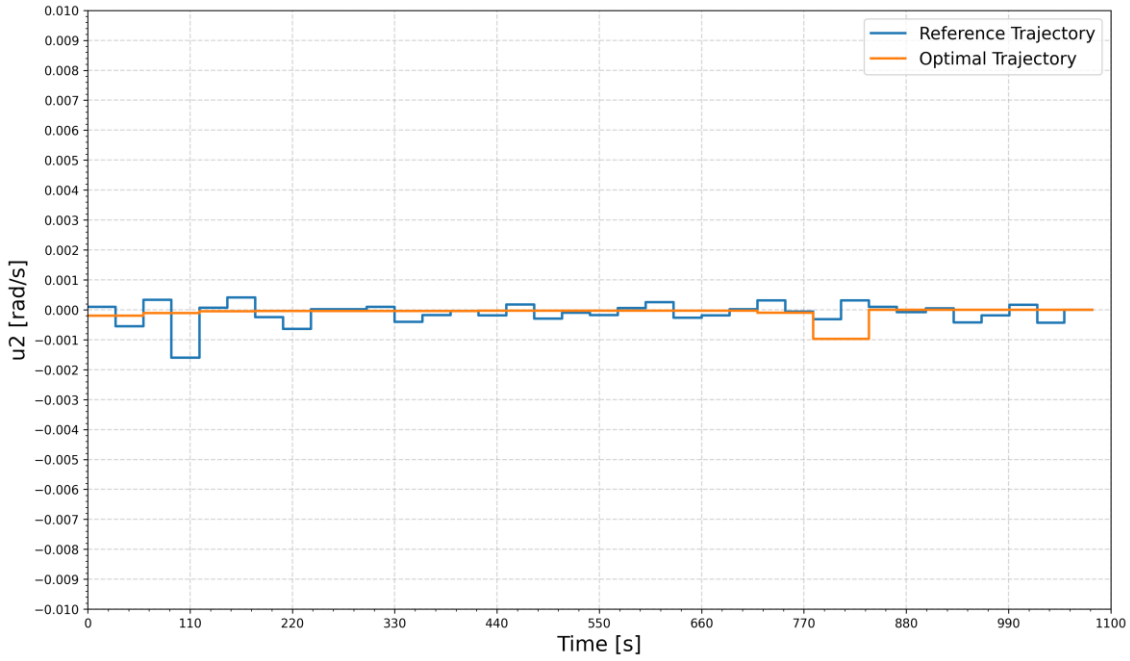


Figure 6.7: Comparison of control u2 vs time of reference and optimal trajectory in climb phase.

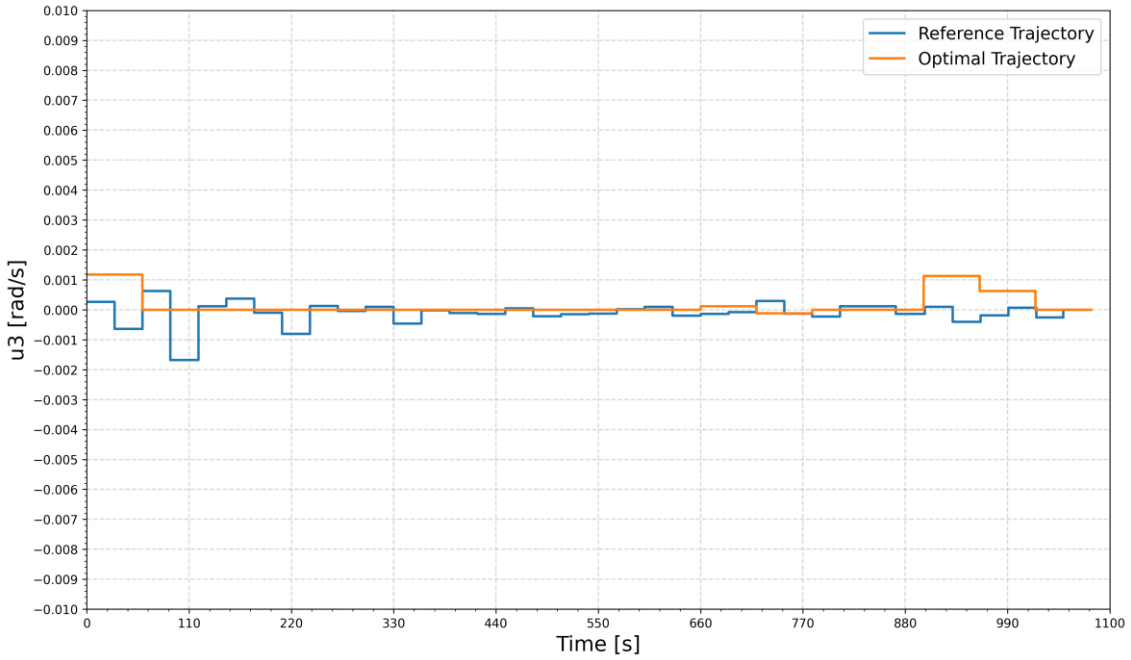


Figure 6.8: Comparison of control u3 vs time of reference and optimal trajectory in climb phase.

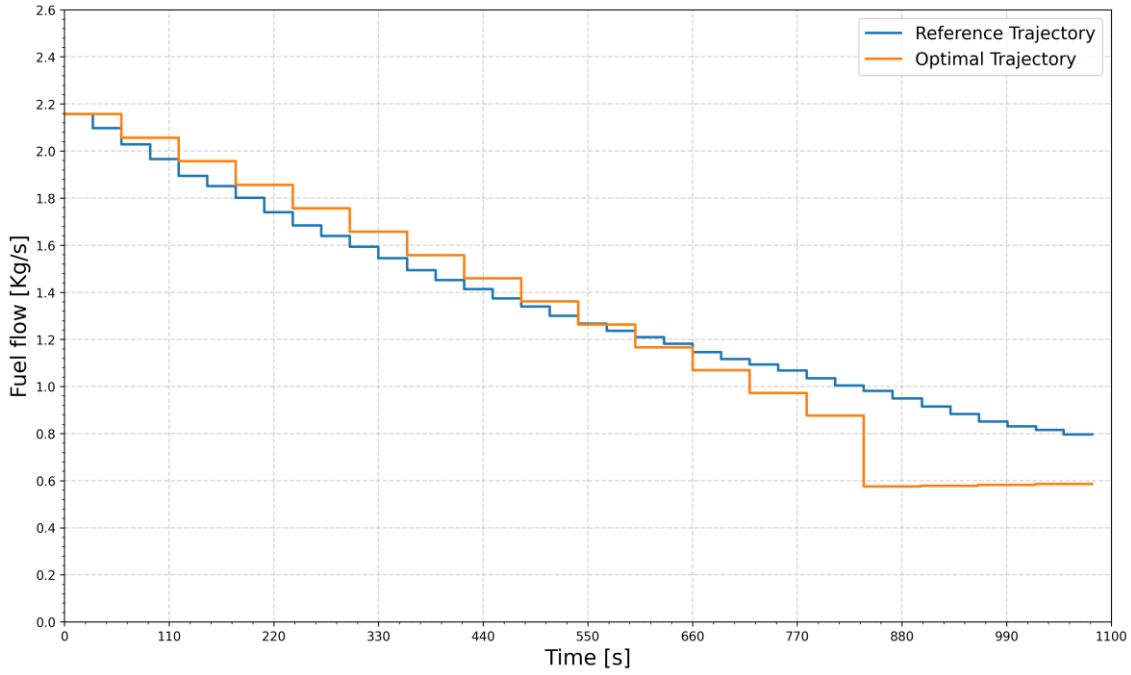


Figure 6.9: Comparison of fuel flow rate vs time of reference and optimal trajectory in climb phase.

Figs. (6.6) – (6.8) represent the time history of controls acceleration u1 in $\left[\mathrm{m/s^2}\right]$, flight path angle rate u2 in $\left[\mathrm{rad/s}\right]$, and heading angle rate u3 in $\left[\mathrm{rad/s}\right]$ of the fuel optimal and reference

trajectories respectively. Fig (6.9) illustrates the fuel flow rate comparison in $[\text{kg/s}]$ between the trajectories. It can be noticed that although the fuel flow rate is higher at the beginning of the optimal trajectory than the reference trajectory. However, as the optimal trajectory climbs higher than the reference trajectory and reaches the ToC earlier it reduces the fuel flow rate, resulting in a reduction of the total fuel consumption in this phase.

Table 6.2: Fuel consumption of reference and optimal trajectory in climb phase.

	Reference Trajectory	**Optimal Trajectory**
Fuel Consumption $[\text{kg}]$	1462.525	1409.175

The fuel consumption of reference and optimal trajectory in the climb phase are shown in table (6.2). The results in the table suggest that the optimal climb trajectory reduces the fuel consumption by 53.35 $[\text{kg}]$, Which is approximately a 3.6% reduction of total climb fuel consumption.

6.4.1.2 Cruise Phase

The second example is a cruise phase of a flight, the time period following the climb phase. The initial and final waypoints of the problem are also set identical to the initial and final waypoints of the cruise phase of a reference commercial flight trajectory. The initial and final 4D waypoints (x, y, h, t) of the cruise phase are shown in table (6.3).

Table 6.3: Initial and final waypoints of the trajectory in cruise phase.

Waypoint	x **[m]**	y **[m]**	h **[m]**	t **[s]**
Initial	4842143.116	-614830.488	11500	0
Final	4437862.141	-11556.358	11550	3840

The comparison between fuel optimal vs reference flight trajectory in the cruise phase of flight is presented in Figs. (6.10) – (6.17). where the orange line represents the time history of the optimal trajectory, and the blue line represents the time history of the reference commercial trajectory.

The three-dimensional (x, y, h) position of the reference and optimal trajectory in the cruise phase of flight is shown in Fig. (6.10). It can be noticed from Figs. (6.10) that cruise altitude of the fuel optimal trajectory is higher than the reference trajectory. Fig. (6.11) shows the true airspeed in $[\text{m/s}]$ of both optimal and reference trajectories, it can also be noticed from Fig. (6.11) that in optimal trajectory the true airspeed is lower than the reference trajectory. This combination of higher cruise altitude and lower cruise true airspeed allow the aircraft to reduce fuel consumption.

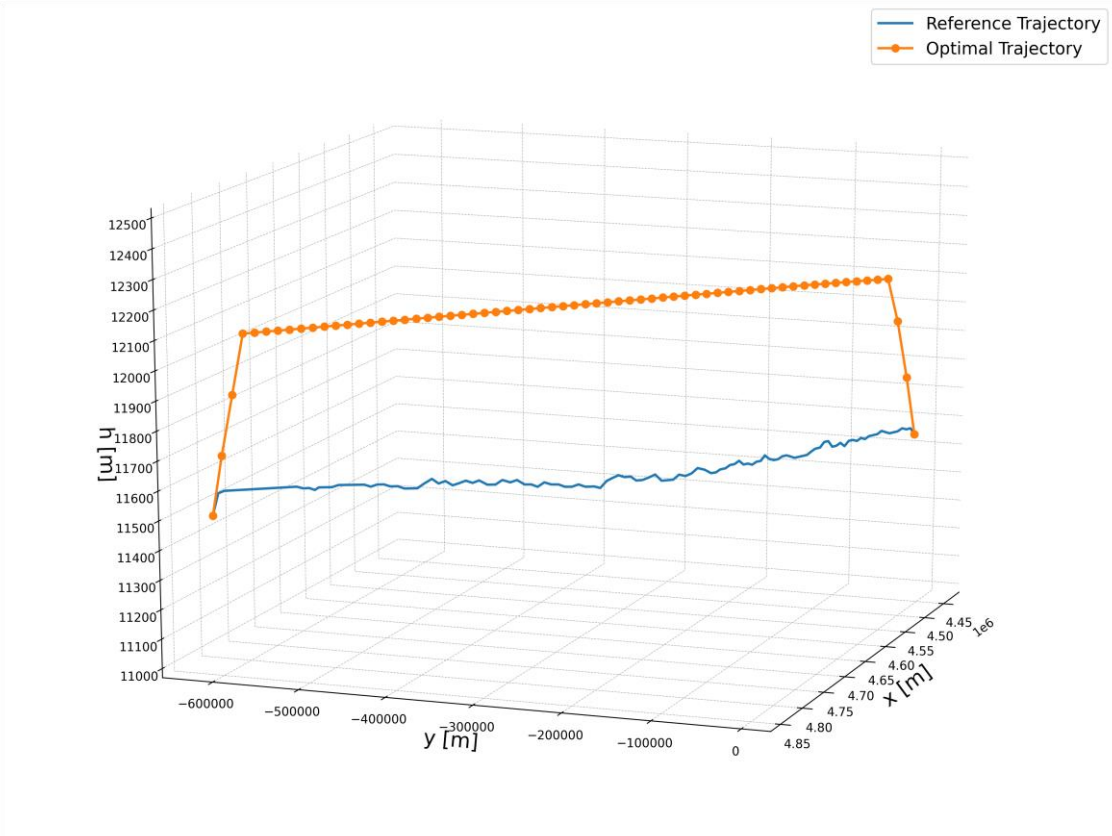


Figure 6.10: Comparison of reference and optimal 3D trajectory in cruise phase.

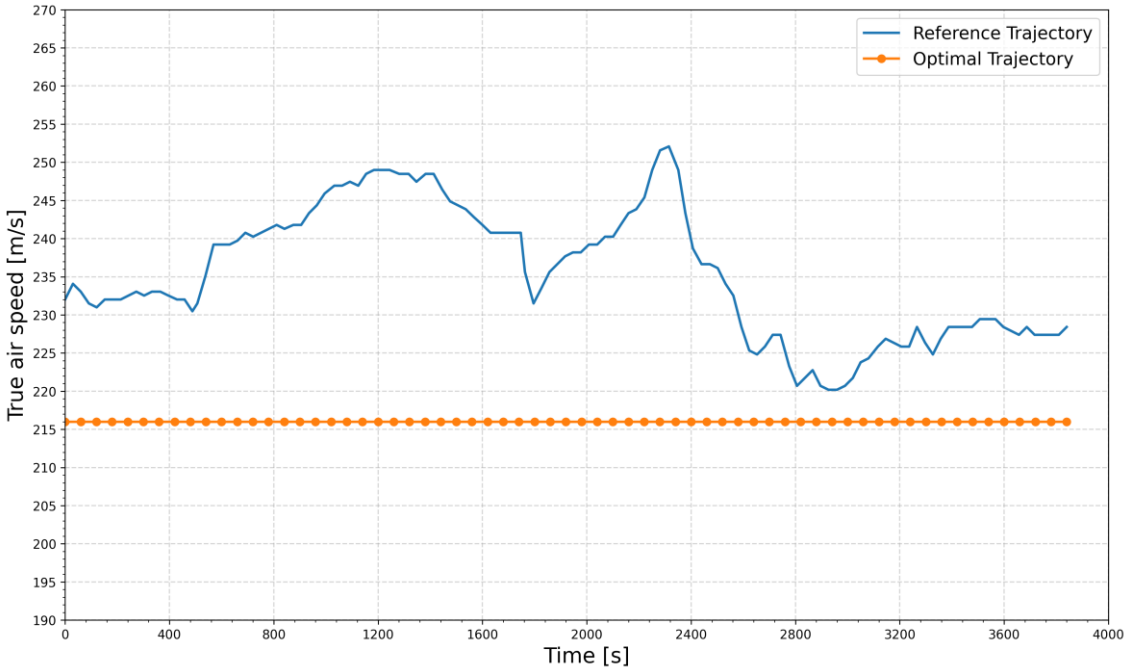


Figure 6.11: Comparison of true airspeed vs time of reference and optimal trajectory in cruise phase.

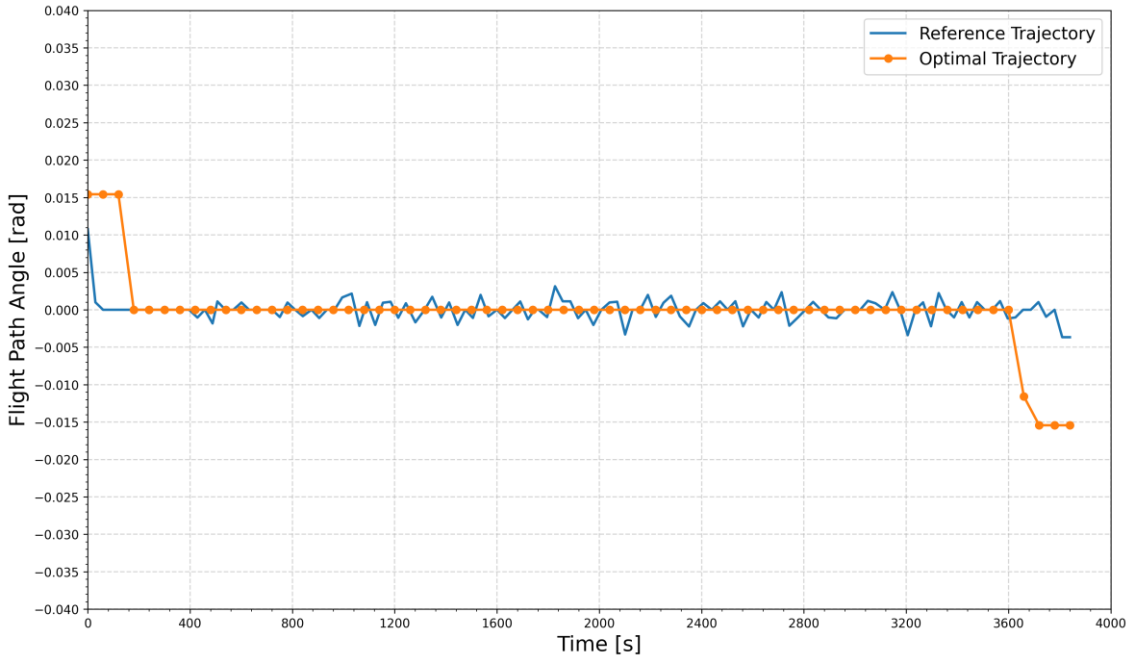


Figure 6.12: Comparison of flight path angle vs time of reference and optimal trajectory in cruise phase.

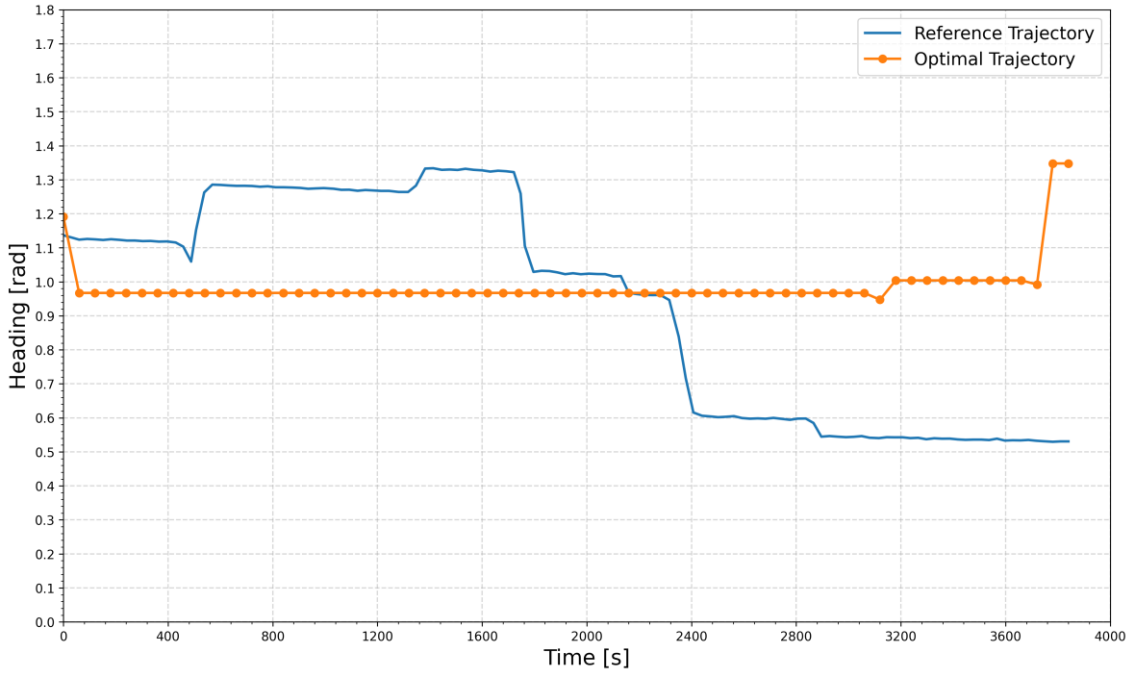


Figure 6.13: Comparison of heading vs time of reference and optimal trajectory in cruise phase.

Fig. (6.12) and (6.13) show the comparison of flight path angle and heading angle in $[\text{rad}]$ for both of the trajectories, as expected the flight path angle is mostly zero in fuel optimal trajectory as the

aircraft flies at the optimum cruise altitude most of the trajectory. In these figures, the flight path angle, and heading angle of the optimal trajectory are within the bounded constraints that were imposed on them.

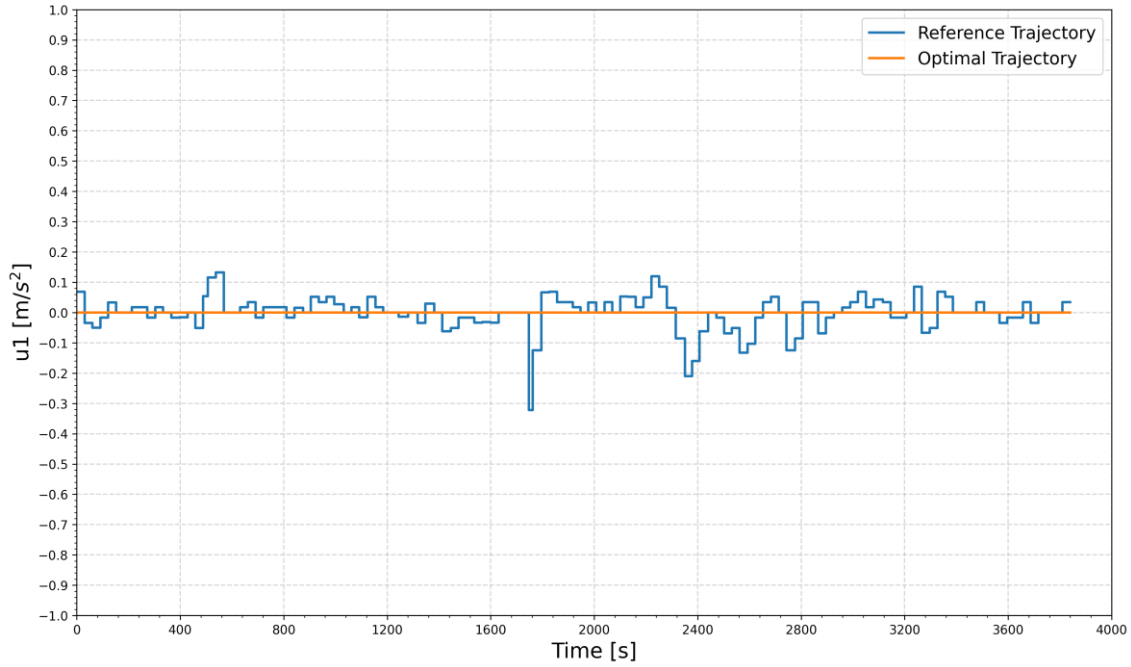


Figure 6.14: Comparison of control u1 vs time of reference and optimal trajectory in cruise phase.

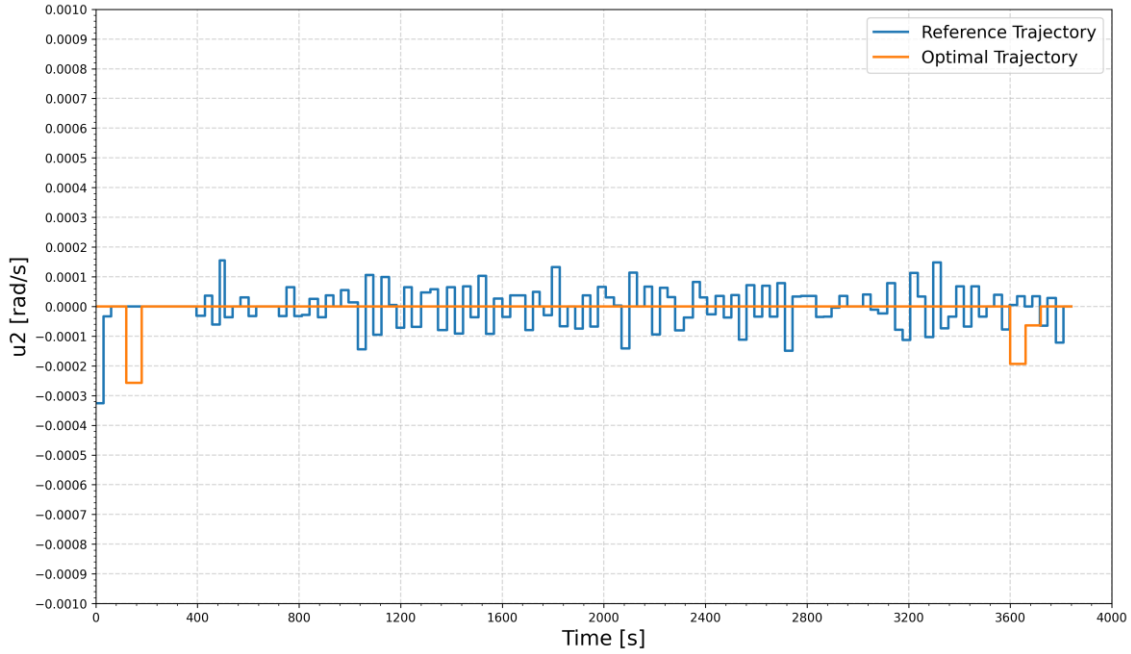


Figure 6.15: Comparison of control u2 vs time of reference and optimal trajectory in cruise phase.

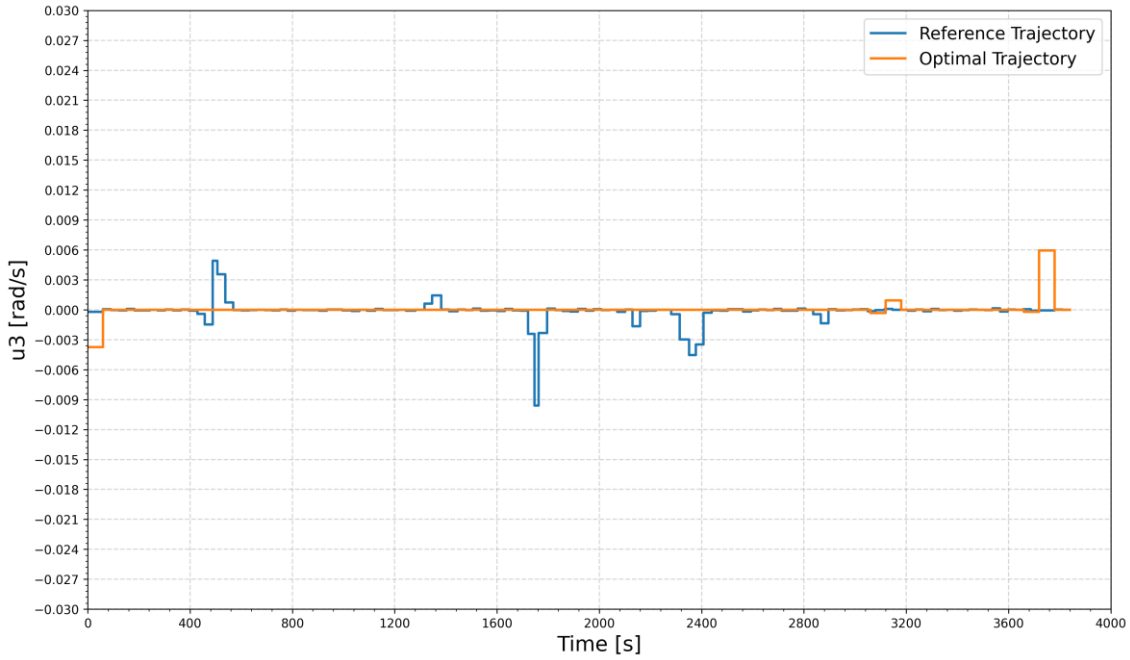


Figure 6.16: Comparison of control u3 vs time of reference and optimal trajectory in cruise phase.

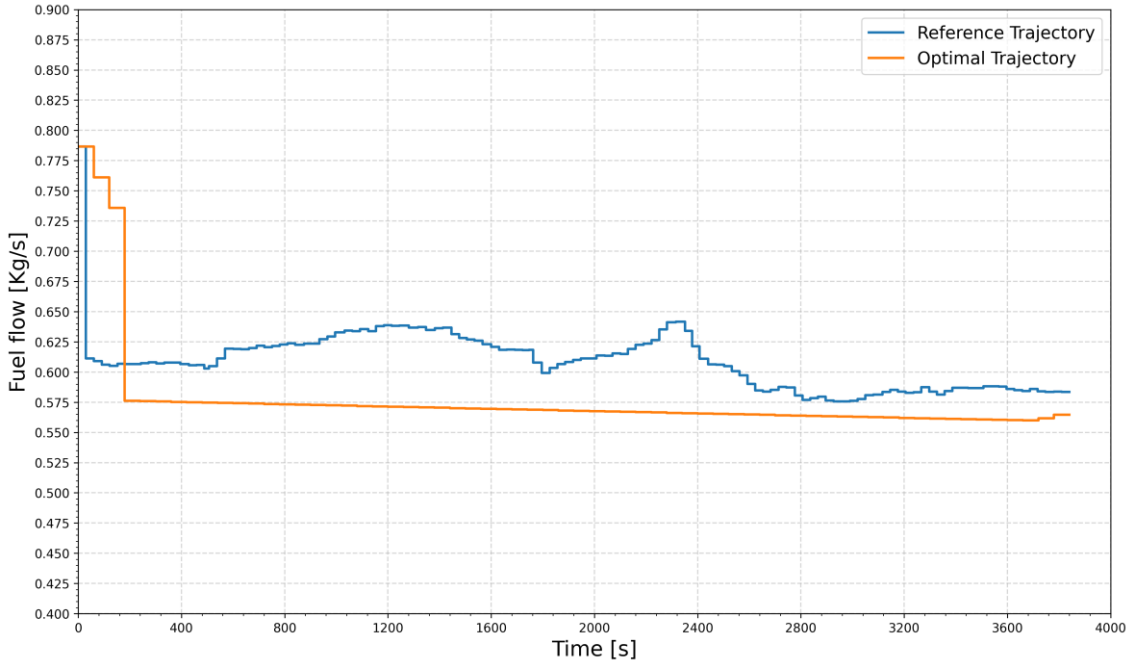


Figure 6.17: Comparison of fuel flow rate vs time of reference and optimal trajectory in cruise phase.

Figs. (6.14) – (6.16) represent the time history of controls acceleration u1 in $\left[\mathrm{m/s^2}\right]$, flight path angle rate u2 in $\left[\mathrm{rad/s}\right]$, and heading angle rate u3 in $\left[\mathrm{rad/s}\right]$ of the fuel optimal and reference

trajectories respectively. Fig. (6.17) shows the fuel flow rate comparison in $[\text{kg/s}]$ between the reference and fuel optimal trajectory. At the beginning of the time history, the fuel flow rate is higher in the fuel optimal trajectory because the aircraft is climbing from the initial cruise altitude to the optimum cruise altitude. Once the optimum cruise altitude is obtained the fuel flow rate is much lower in the fuel optimal trajectory than the reference trajectory for the rest of the time history.

Table 6.4: Fuel consumption of reference and optimal trajectory in cruise phase.

	Reference Trajectory	**Optimal Trajectory**
Fuel Consumption [kg]	2338.323	2215.001

Table (6.4) shows the results of fuel consumption for both the trajectory, the total fuel consumption by the aircraft for the reference cruise phase trajectory is $2339.657\ [\text{kg}]$, and the fuel optimal trajectory is $2215.038\,[\text{kg}]$. This implies that by flying the optimal trajectory approximately $123.322\ [\text{kg}]$ (5.3%) of fuel can be saved than the reference trajectory.

6.4.1.3 Descent Phase

The third example is a descent phase of a flight, the time period following the cruise phase. Like the other cases, it is assumed that the optimal trajectory starts at the initial waypoint and ends at the final waypoint of a reference commercial descent flight trajectory. The initial and final 4D waypoints (x, y, h, t) of the descent phase are shown in table (6.5).

Table 6.5: Initial and final waypoints of the trajectory in descent phase.

Waypoint	***x* [m]**	***y* [m]**	***h* [m]**	***t* [s]**
Initial	4437862.141	-11556.358	11550	0
Final	4193076.371	227771.503	1550	2340

The comparison between fuel optimal vs reference flight trajectory in the descent phase of flight is presented in Figs. (6.18) – (6.25). where the orange line represents the optimal trajectory, and the blue line represents the time history of the reference commercial trajectory.

The three-dimensional (x, y, h) position of reference and fuel optimal trajectory is shown in Fig. (6.18). The figure suggests that the top of descent (ToD) occurs later in the optimal trajectory and the aircraft descend continuously to reduce the fuel consumption, unlike the reference trajectory where the descend occurs in the conventional step-down procedure, which results in more fuel burn.

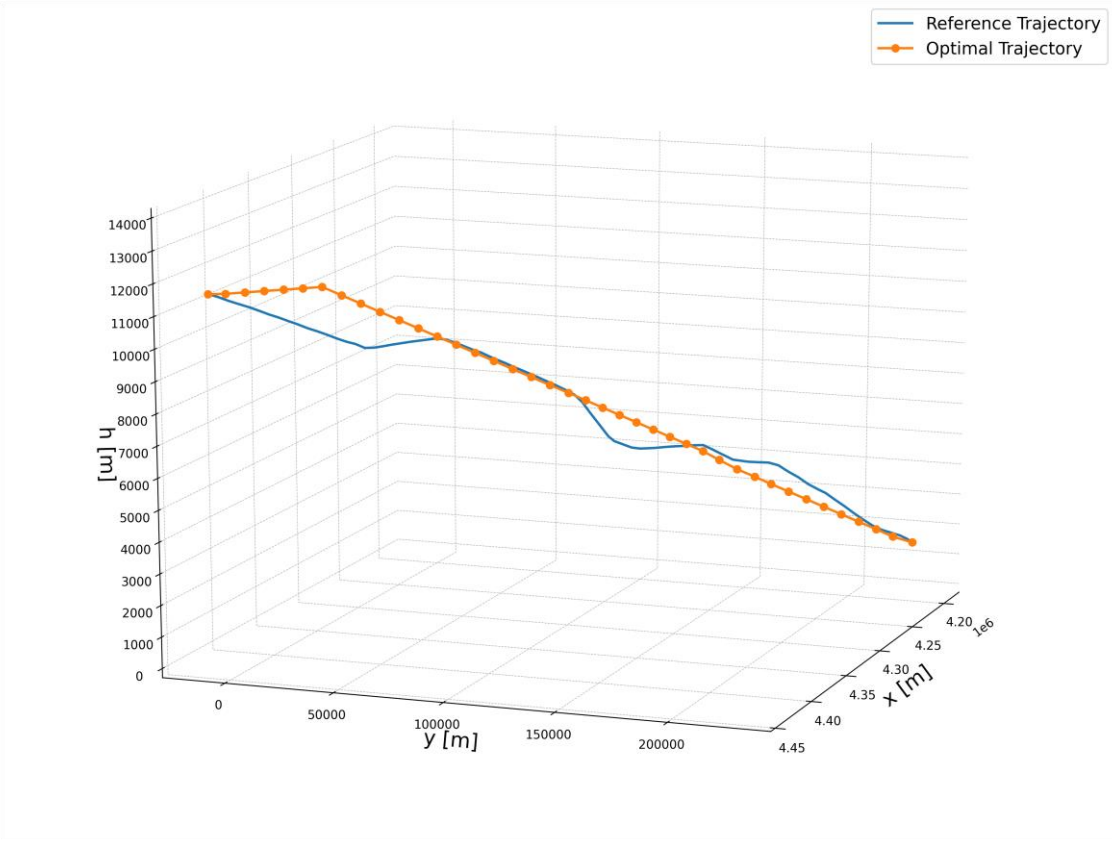


Figure 6.18: Comparison of reference and optimal 3D trajectory in descent phase.

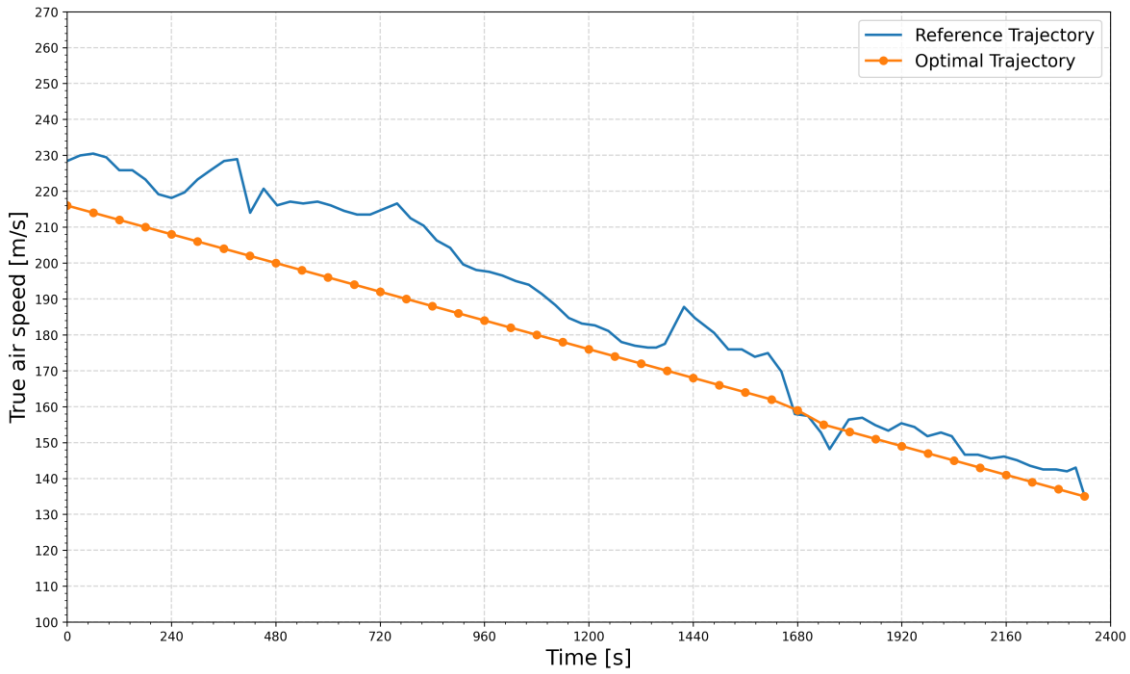


Figure 6.19: Comparison of true airspeed vs time of reference and optimal trajectory in descent phase.

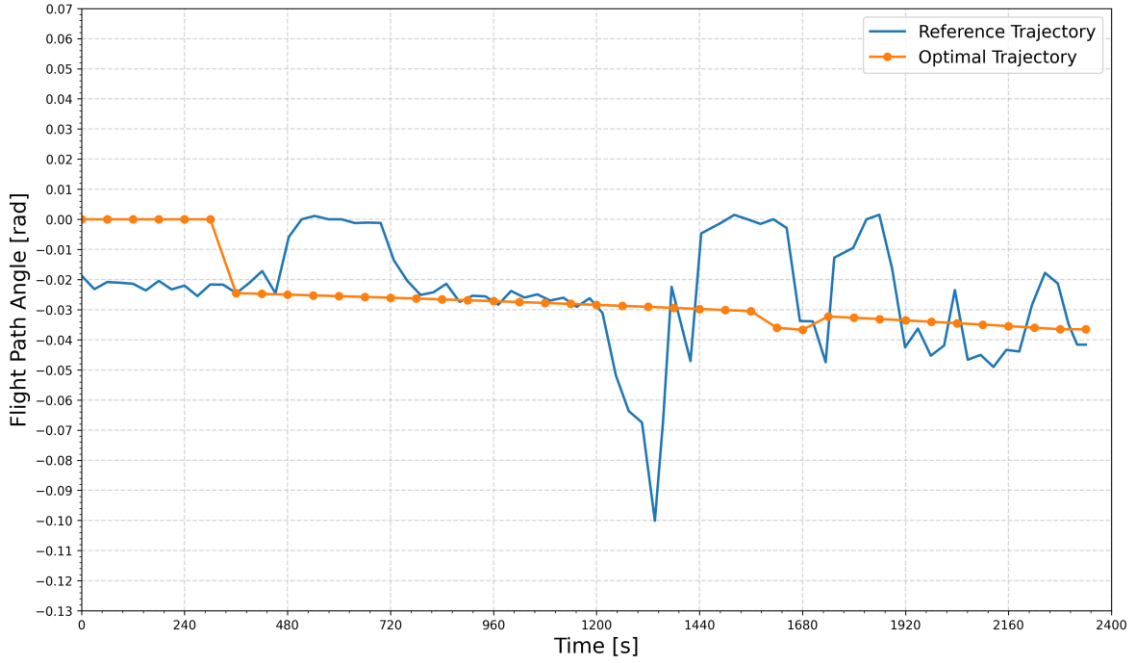


Figure 6.20: Comparison of flight path angle vs time of reference and optimal trajectory in descent phase.

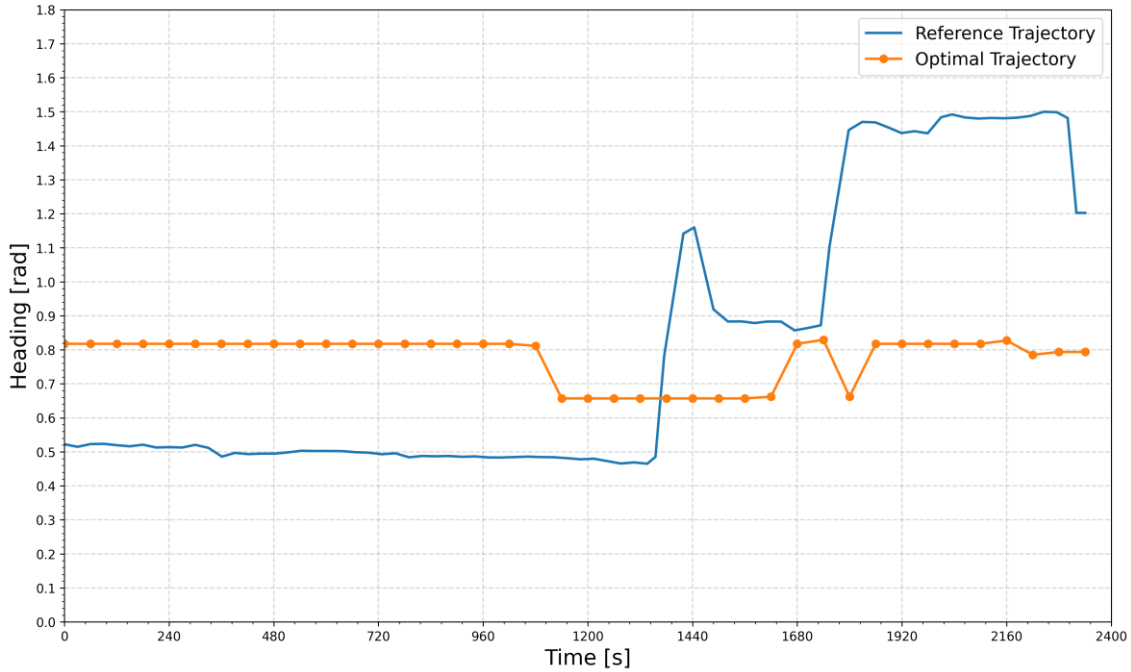


Figure 6.21: Comparison of heading vs time of reference and optimal trajectory in descent phase.

Fig. (6.19) shows the comparison of true airspeed in $[\mathrm{m/s}]$, Fig. (6.20) shows the time history of the flight path angle in $[\mathrm{rad}]$ and Fig. (6.21) shows the heading angle in $[\mathrm{rad}]$ between the

reference and fuel optimal trajectory. In these figures, the true airspeed, flight path angle, and heading angle of the optimal trajectory are within the bounded constraints that were imposed on them.

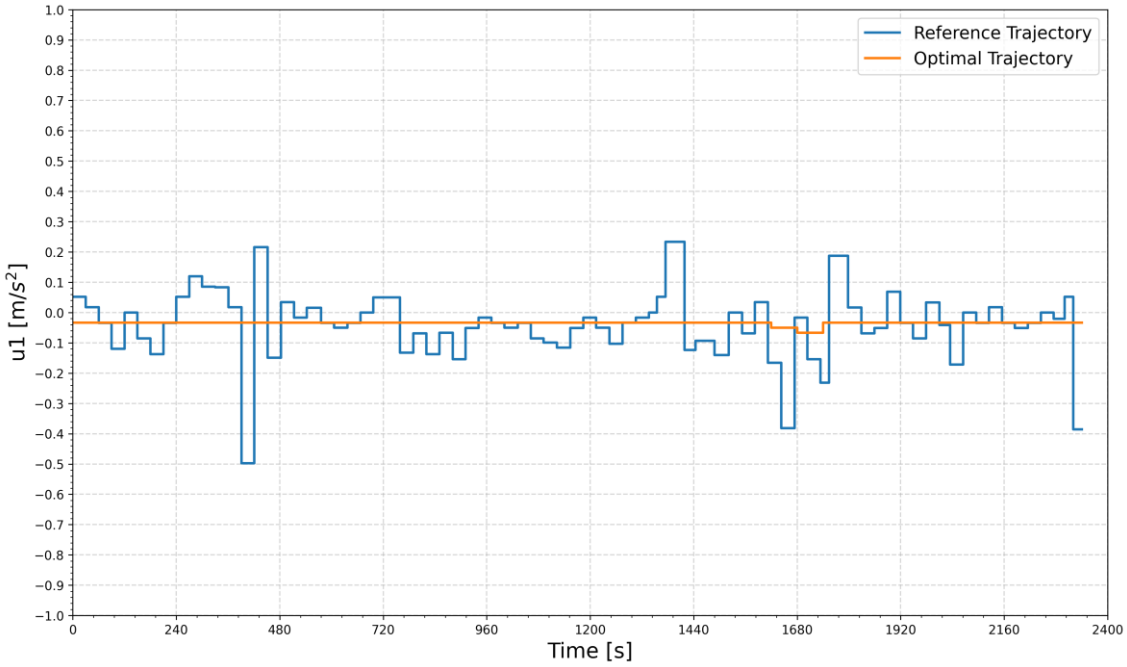


Figure 6.22: Comparison of control u1 vs time of reference and optimal trajectory in descent phase.

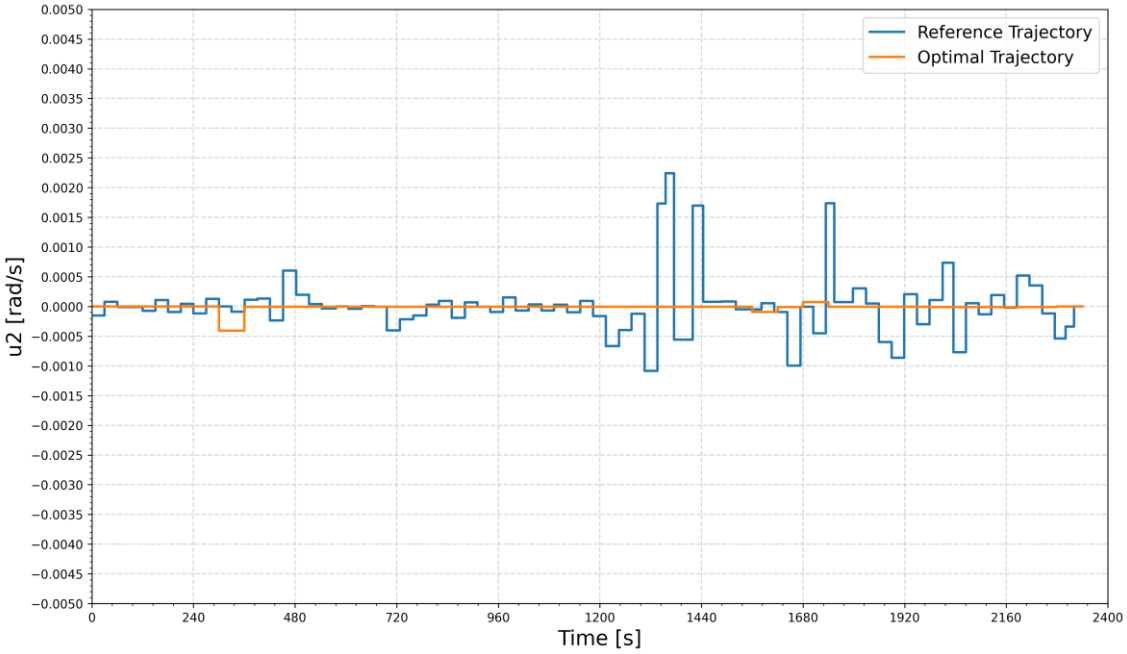


Figure 6.23: Comparison of control u2 vs time of reference and optimal trajectory in descent phase.

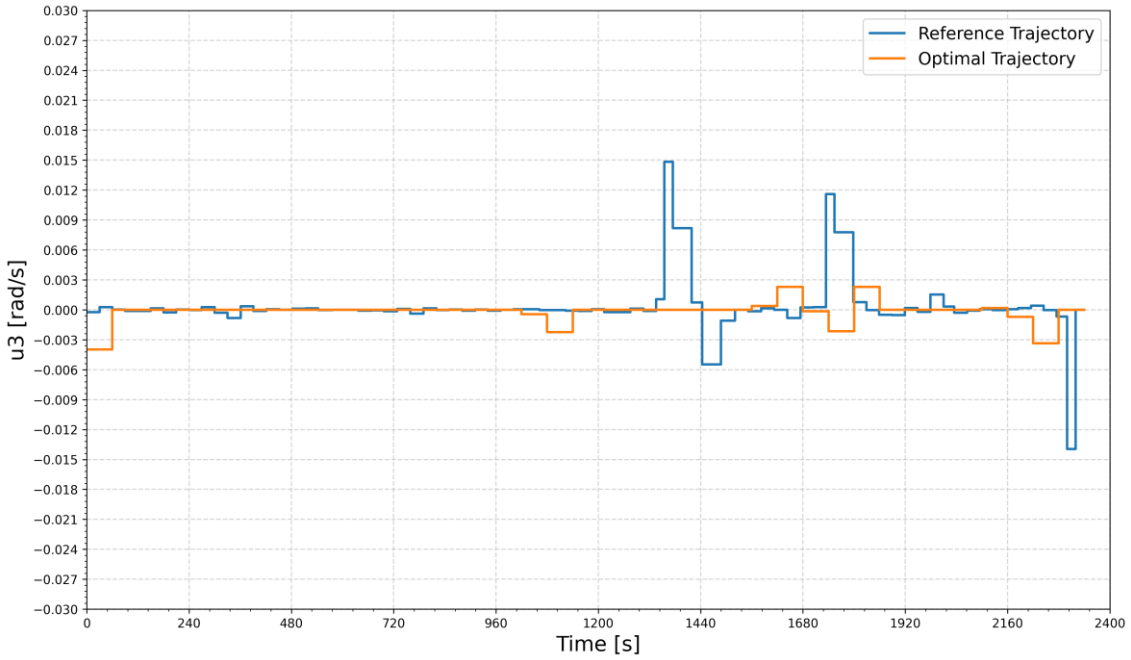


Figure 6.24: Comparison of control u3 vs time of reference and optimal trajectory in descent phase.

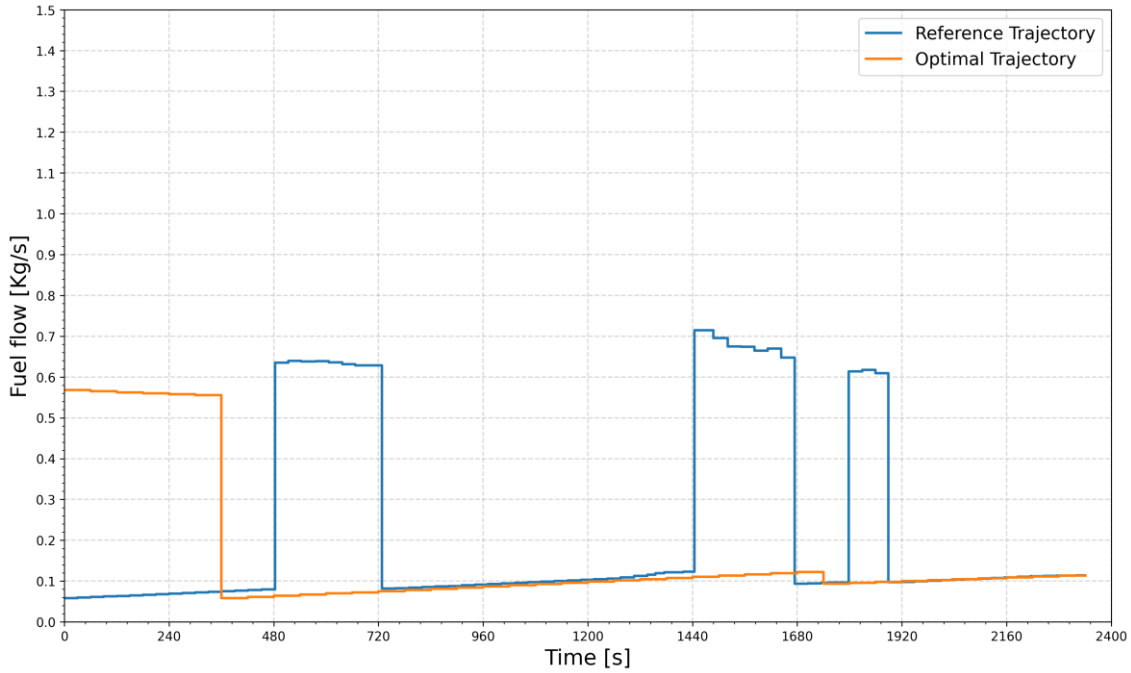


Figure 6.25: Comparison of fuel flow rate vs time of reference and optimal trajectory in descent phase.

Figs. (6.22) – (6.24) represent the time history of controls acceleration u1 in $\left[\mathrm{m/s^2}\right]$, flight path angle rate u2 in $\left[\mathrm{rad/s}\right]$, and heading angle rate u3 in $\left[\mathrm{rad/s}\right]$ of the fuel optimal and reference

trajectories respectively. the fuel flow rate comparison between the reference and fuel optimal trajectory is presented in Fig. (6.25). It can be noticed from Fig. (6.25) that the fuel flow rate of the reference trajectory spiked on several occasions, this occurred since the reference trajectory used the inefficient step-down descent procedure.

Table 6.6: Fuel consumption of reference and optimal trajectory in descent phase.

	Reference Trajectory	**Optimal Trajectory**
Fuel Consumption [kg]	531.391	388.228

Table (6.6) compares the fuel consumption of reference and optimal trajectory in descent phase. The results suggest that by flying the optimal trajectory a 143.163 [kg] reduction of fuel consumption can be achieved, which is equivalent to 26.9 % of fuel consumption reduction than the reference trajectory.

6.4.1.4 Global Trajectory

This example considered the global trajectory, which consists of all three climb, cruise, and descent phases of flight. The initial and final waypoints of the problem are set identical to the initial and final waypoints of the reference commercial flight trajectory. The initial and final 4D waypoints (x, y, h, t) are shown in table (6.7).

Table 6.7: Initial and final waypoints of the global trajectory.

Waypoint	***x* [m]**	***y* [m]**	***h* [m]**	***t* [s]**
Initial	4912063.702	-784235.333	1050	0
Final	4193076.371	227771.503	1550	7260

Figs. (6.26) – (6.33) show the comparison between fuel optimal vs reference commercial global trajectory, where the orange line represents the time history of the optimal trajectory, and the blue line represents the time history of reference commercial trajectory as exported from the website Flightaware.

Fig. (6.26) shows the three-dimensional (x, y, h) position of the reference and optimal global trajectory, where the aircraft reaches the Top of Climb (ToC) at 1020 [s] and the Top of Descent (ToD) at 5100 [s] in optimal trajectory. The cruise altitude of the optimal trajectory is at 12500 [m] with a constant speed of 216 [m/s].

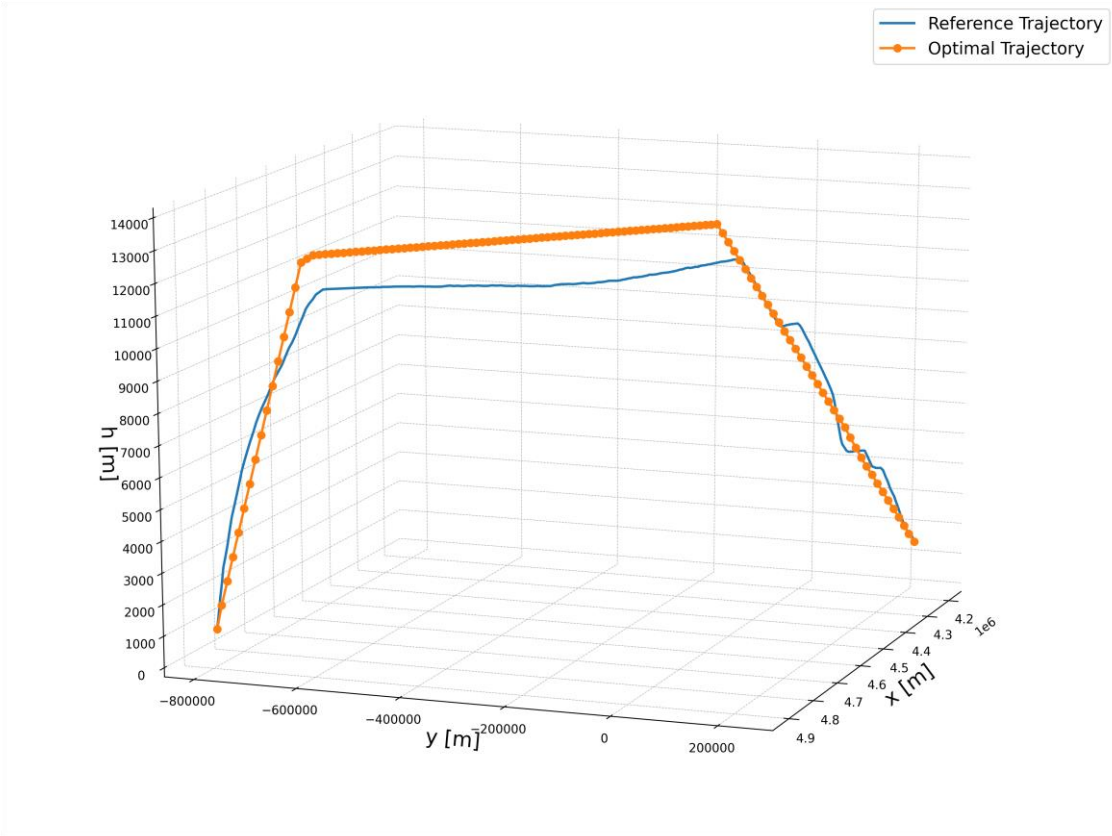


Figure 6.26: Comparison of reference and optimal global 3D trajectory.

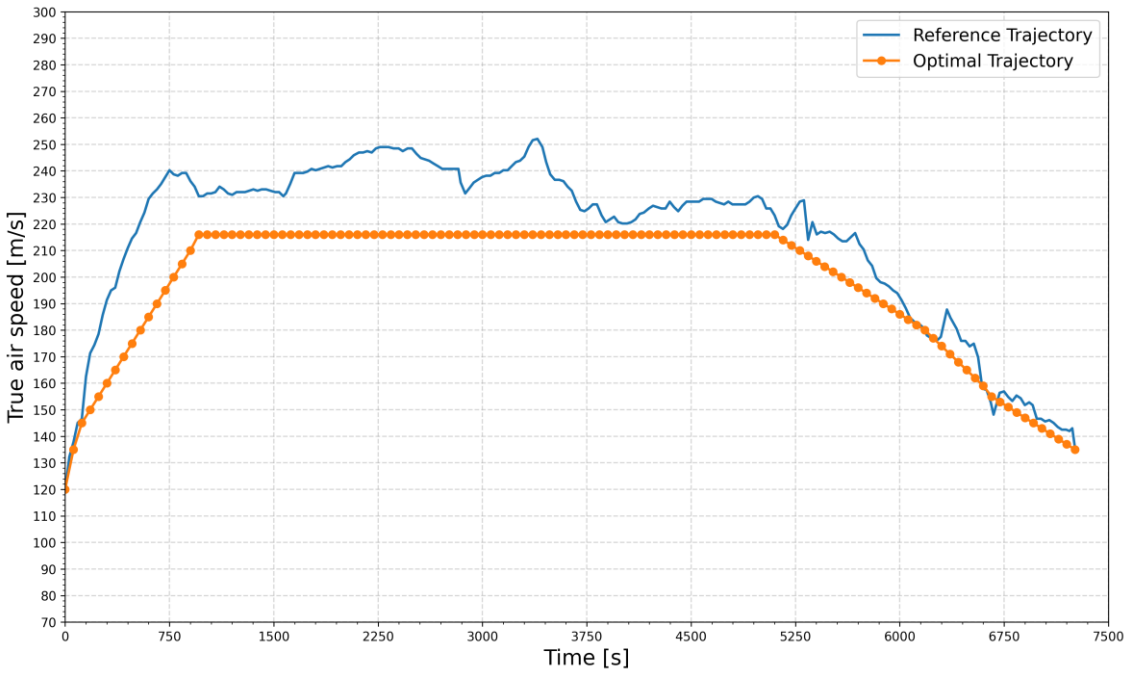


Figure 6.27: Comparison of true airspeed vs time of reference and optimal global trajectory.

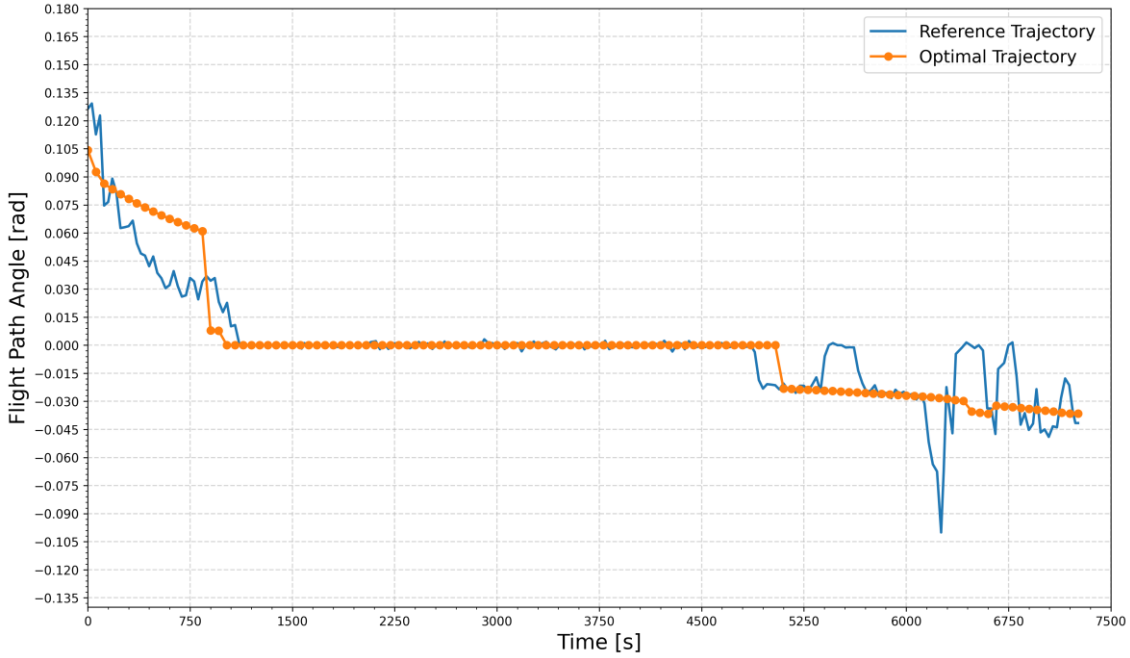


Figure 6.28: Comparison of flight path angle vs time of reference and optimal global trajectory.

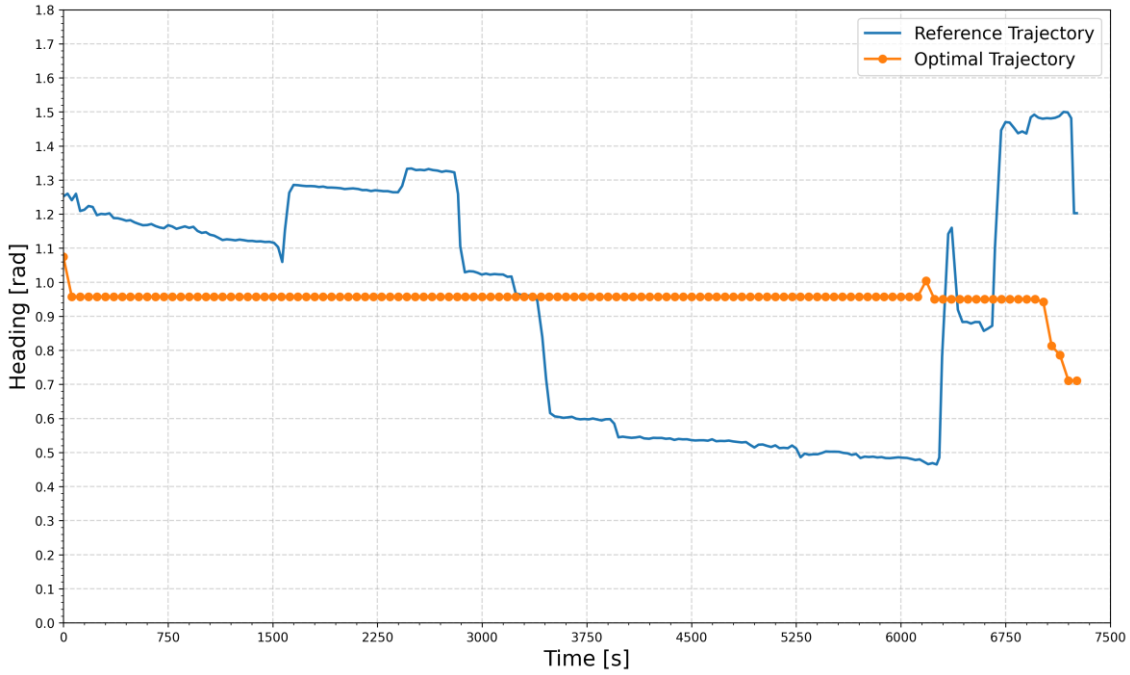


Figure 6.29: Comparison of Heading vs time of reference and optimal global trajectory.

Fig. (6.27) shows the comparison of true airspeed in $[\mathrm{m/s}]$, Fig. (6.28) shows the time history of the flight path angle in $[\mathrm{rad}]$and Fig. (6.29) shows the heading angle in $[\mathrm{rad}]$ between the

reference and fuel optimal global trajectory. In these figures, the true airspeed, flight path angle, and heading angle of the optimal trajectory are within the bounded constraints that were imposed on them.

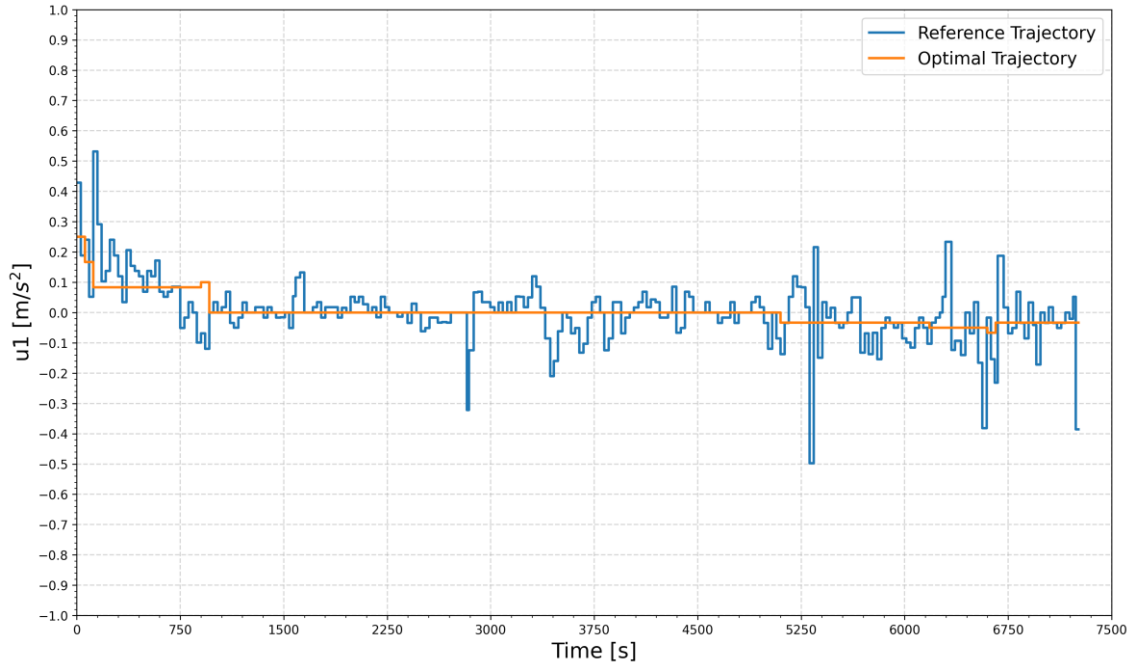


Figure 6.30: Comparison of control u1 vs time of reference and optimal global trajectory.

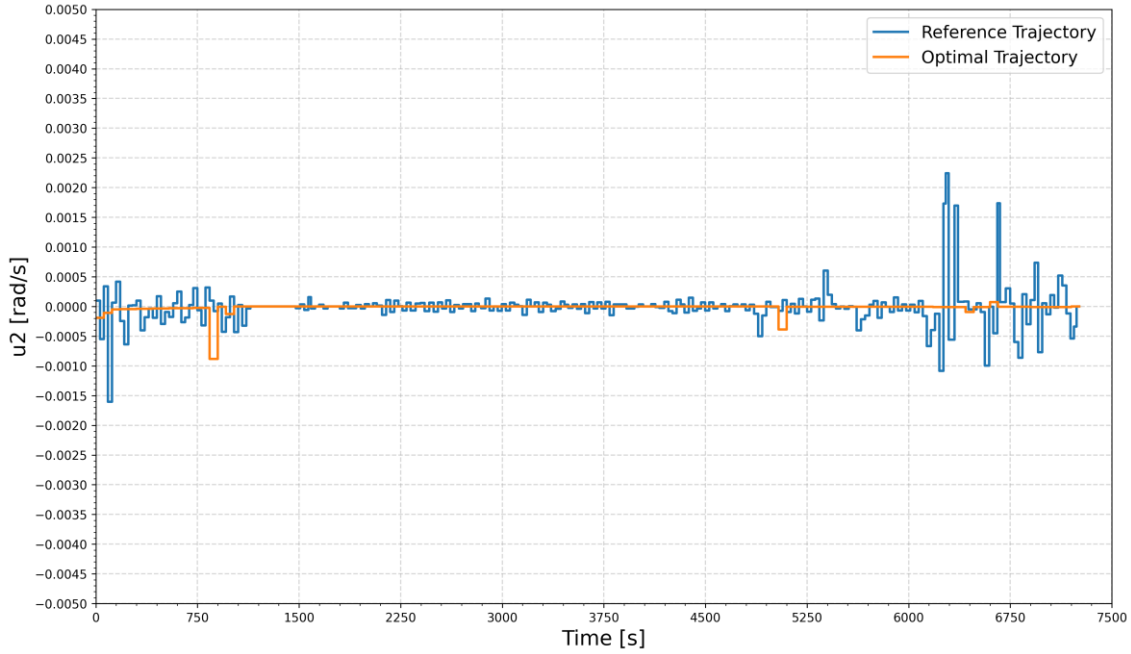


Figure 6.31: Comparison of control u2 vs time of reference and optimal global trajectory.

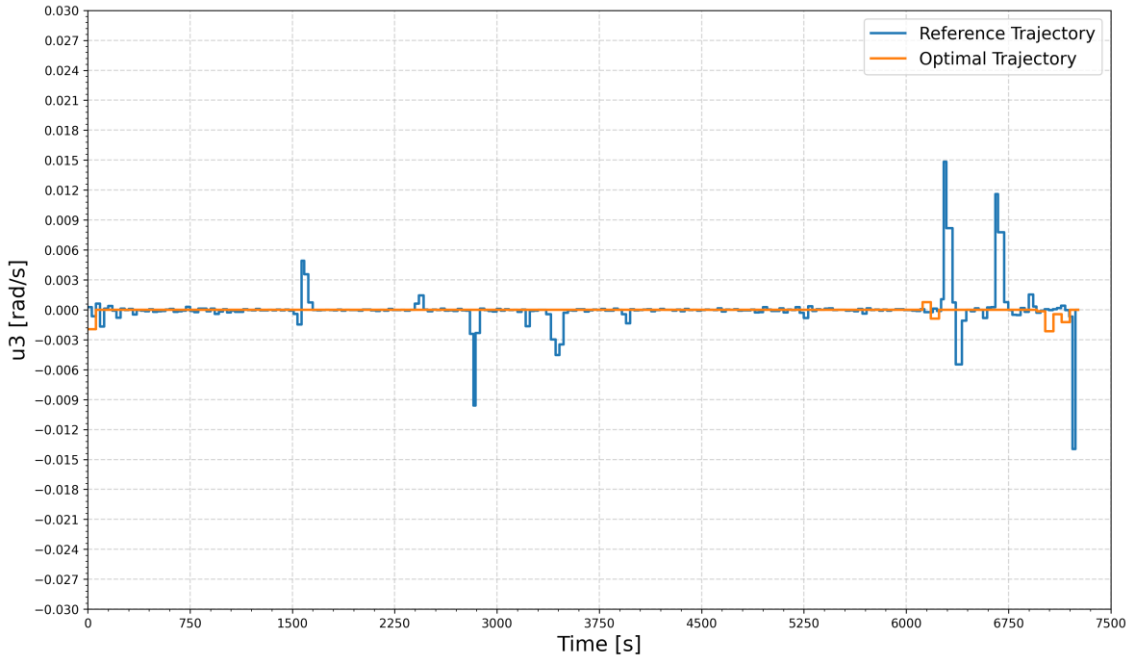


Figure 6.32: Comparison of control u3 vs time of reference and optimal global trajectory.

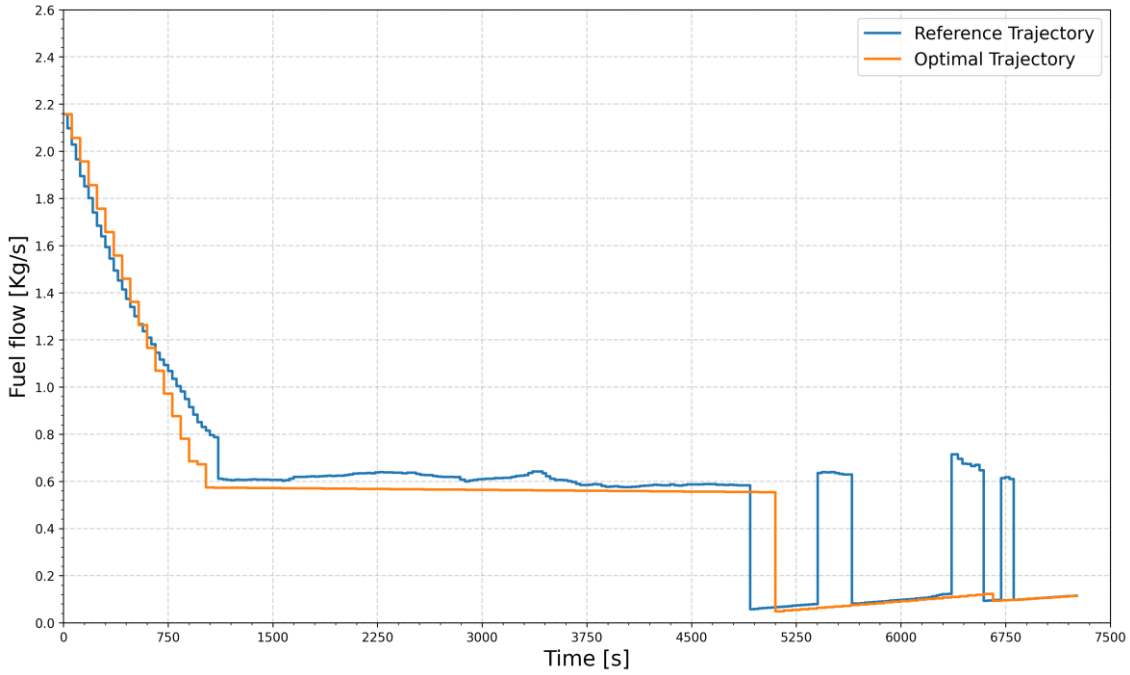


Figure 6.33: Comparison of fuel flow rate vs time of reference and optimal global trajectory.

Figs. (6.30) – (6.32) represent respectively the time history of controls acceleration u1 in $\left[\mathrm{m/s^2}\right]$, flight path angle rate u2 in $\left[\mathrm{rad/s}\right]$, and heading angle rate u3 in $\left[\mathrm{rad/s}\right]$ of the reference and fuel

optimal global trajectories. Fig (6.33) shows the aircraft fuel flow rate comparison between the reference and fuel optimal global trajectories. Unlike the reference trajectory, in the fuel optimal trajectory, the ToC occurs at a higher altitude, then the aircraft continues for the cruise phase at the same altitude to save fuel, as the fuel flow rate is lower in this scenario than the reference trajectory. Later on, the optimal trajectory chooses a continuous descent process instead of the step-down descent procedure to save a great deal of fuel.

Table 6.8: Fuel consumption of reference and optimal global trajectory.

	Reference Trajectory	**Optimal Trajectory**
Fuel Consumption $[\mathbf{kg}]$	4332.239	3892.913

Table (6.8) compares the fuel consumption of the reference and optimal global trajectory. The results suggest that a 439.326 $[\mathrm{kg}]$ reduction, which is approximately a 10.1% reduction of aircraft fuel consumption by flying the optimal trajectory.

6.4.2 Emissions Optimal Trajectory

In this second case study, a flight from Lisbon to Munich and a twinjet aircraft was considered to analyze the emissions optimal trajectories. In this study, the reference mass of this aircraft was considered as the take-off weight, which is 64,000 kg and it is also assumed there is no wind condition. The performance operational data of the aircraft is provided in Annex A.2. In all four examples, to determine the potential benefit of reduction of aircraft emissions, the obtained emissions optimal trajectories were compared with a commercial flight trajectory for the same route.

The constraints of the case study have been selected according to the reference commercial flight trajectory for the same route as follow:

Altitude $[\mathrm{m}]$	:	$2000 \le h(t) \le 11850$
True airspeed $[\mathrm{m/s}]$	:	$135 \le V(t) \le 250$
Flight Path Angle $[\mathrm{rad}]$	:	$-0.06 \le \gamma(t) \le 0.15$
Heading Angle $[\mathrm{rad}]$	:	$0.8 \le \psi(\mathrm{t}) \le 1.6$

In all four examples of this case study, the interval between the states grid points (x, y, h) are $\Delta x = 1000$ [m], $\Delta y = 1000$ [m], $\Delta h = 50$ [m], and the interval between the stages is $\Delta t = 60$ [s].

6.4.2.1 Climb Phase

The first example of this case study is a climb phase of flight. The initial and final waypoints of the problem are set identical to the initial and final waypoints of the climb phase of the reference

commercial flight trajectory. Thus, both the optimal and reference trajectory start and end at the same waypoints. The initial and final 4D waypoints (x, y, h, t) of the problem are shown in table (6.9).

Table 6.9: Initial and final waypoints of the trajectory in climb phase.

Waypoint	x [m]	y [m]	h [m]	t [s]
Initial	4912599.596	-781419.581	2100	0
Final	4865492.37	-663479.172	11500	780

Figs. (6.34) – (6.41) show the comparison between emissions optimal vs reference commercial trajectory in the climb phase of flight, where the orange line represents the time history of the optimal trajectory, and the blue line represents the time history of reference commercial trajectory as exported from the website Flightaware.

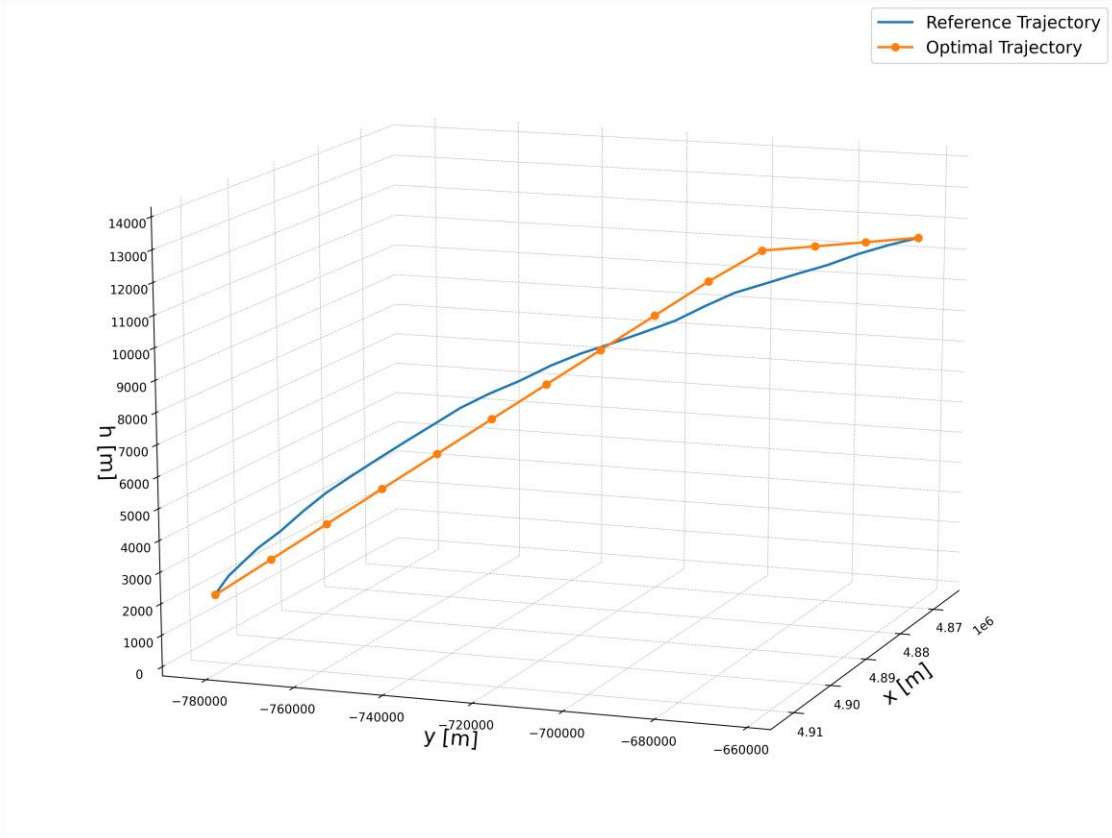


Figure 6.34: Comparison of reference and optimal 3D trajectory in climb phase.

Fig. (6.34) shows the three-dimensional (x, y, h) position of the reference and optimal trajectory in the climb phase of flight. In the optimal trajectory, the Top of Climb (ToC) is achieved earlier than the reference trajectory which allows it to reduce the aircraft emissions. Fig. (6.35) shows the comparison of true airspeed in $[\mathrm{m/s}]$, Fig. (6.36) shows the time history of the flight path

angle in [rad] and Fig. (6.37) shows the heading angle in [rad] between the reference and emissions optimal trajectory. In these figures, the true airspeed, flight path angle, and heading angle of the optimal trajectory are within the bounded constraints that were imposed on them.

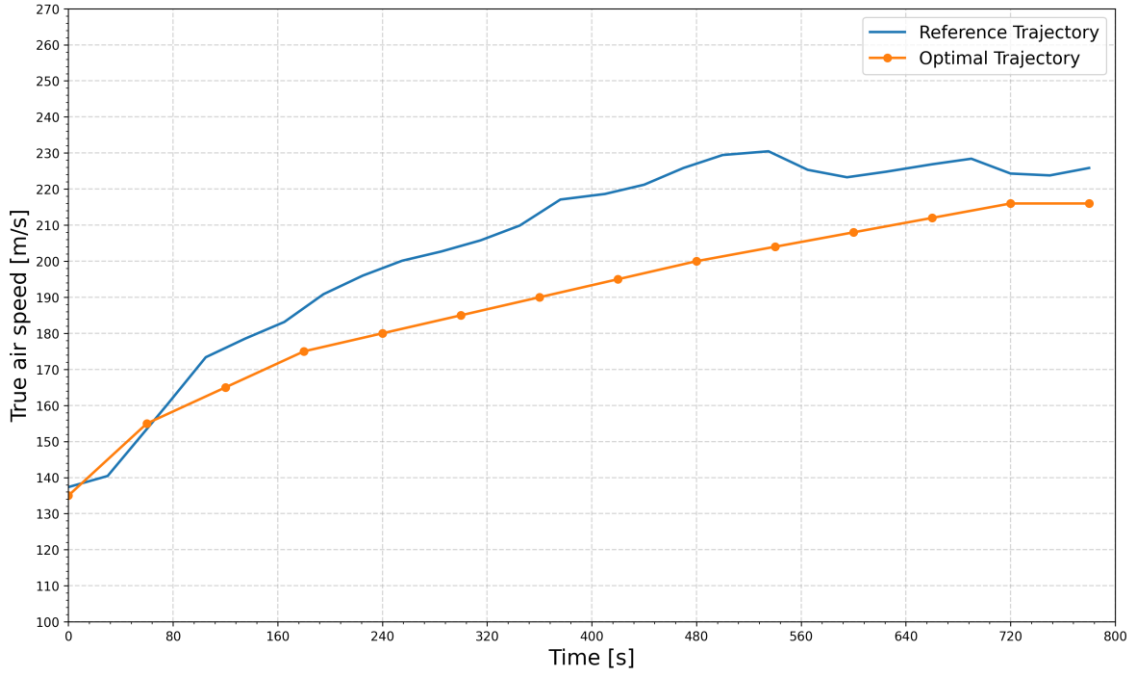


Figure 6.35: Comparison of true airspeed vs time of reference and optimal trajectory in climb phase.

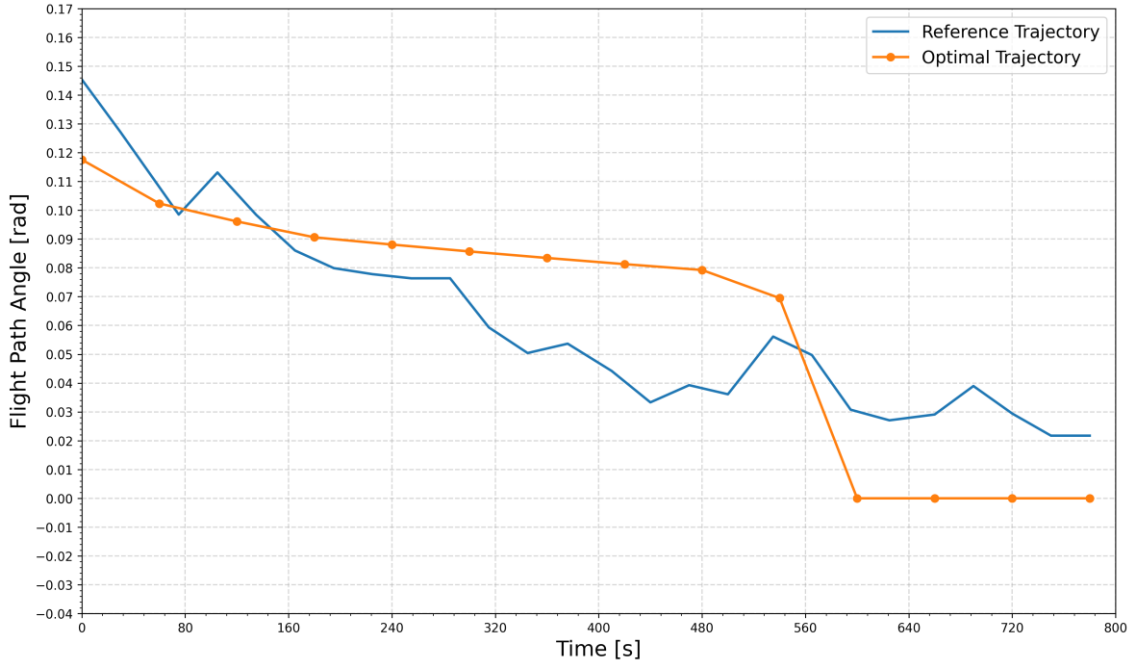


Figure 6.36: Comparison of flight path angle vs time of reference and optimal trajectory in climb phase.

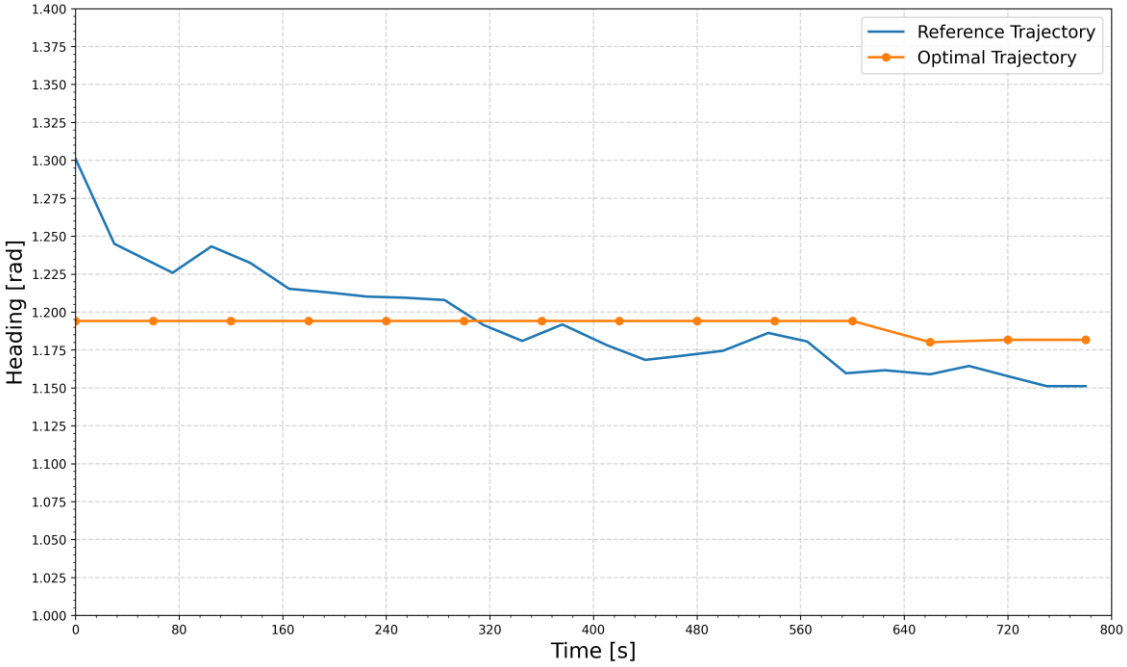


Figure 6.37: Comparison of Heading vs time of reference and optimal trajectory in climb phase.

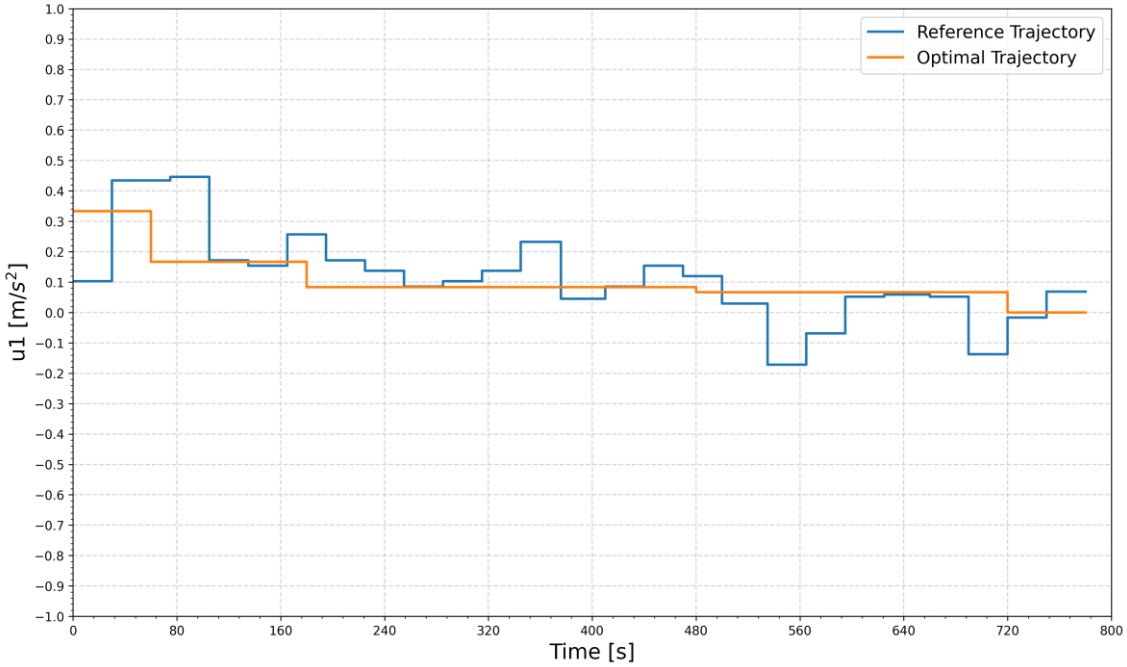


Figure 6.38: Comparison of control u1 vs time of reference and optimal trajectory in climb phase.

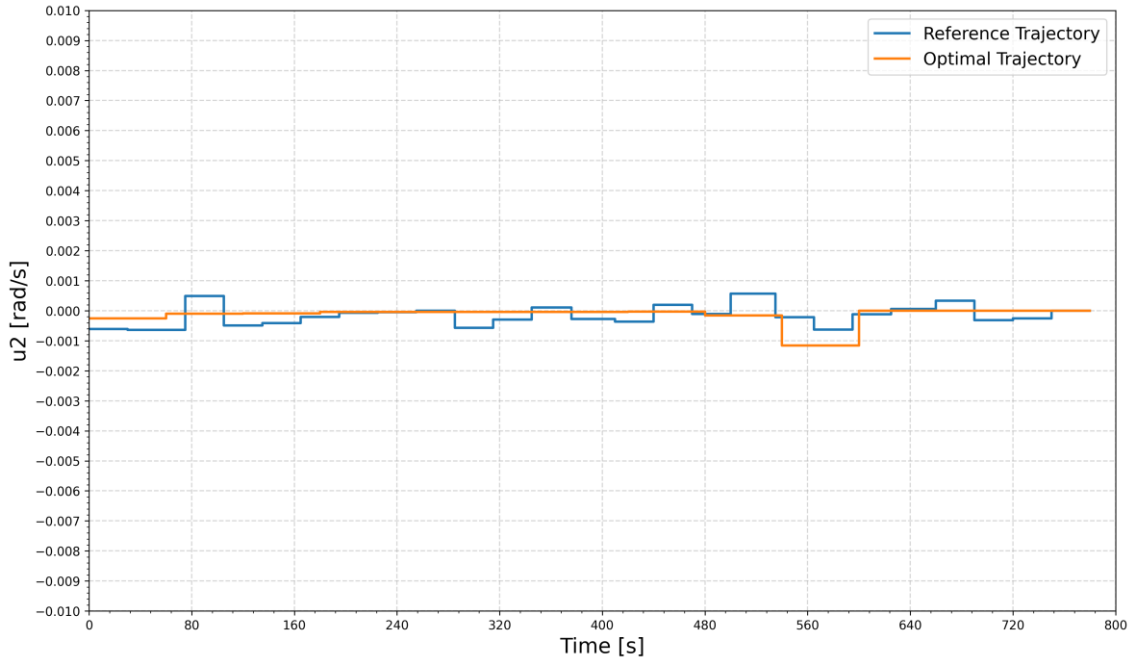


Figure 6.39: Comparison of control u2 vs time of reference and optimal trajectory in climb phase.

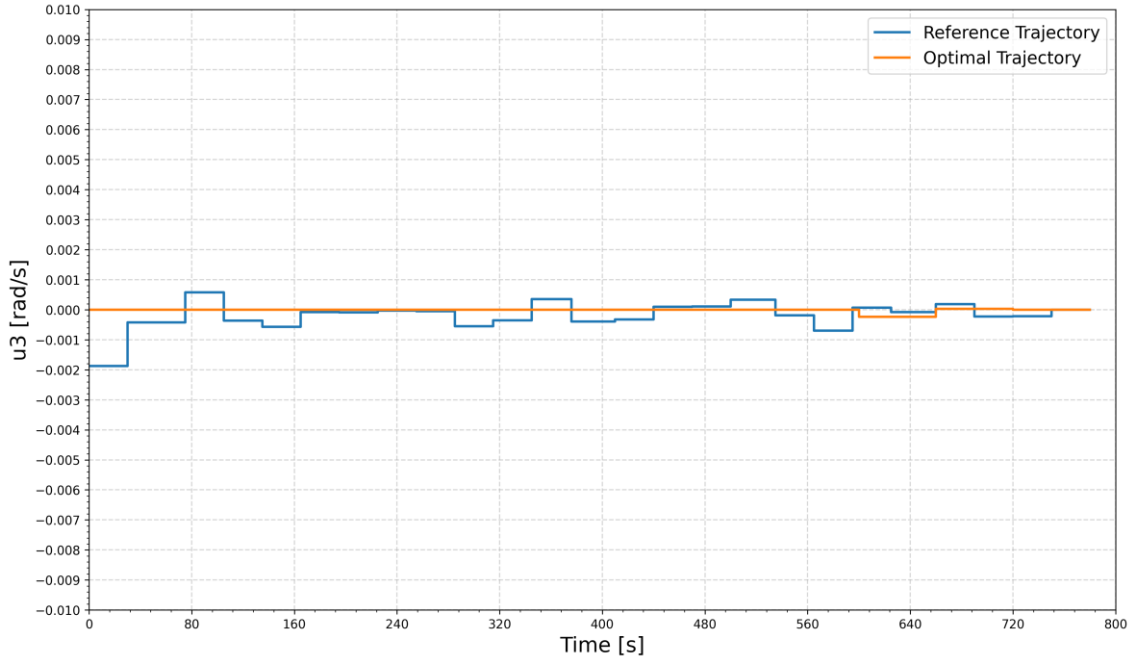


Figure 6.40: Comparison of control u3 vs time of reference and optimal trajectory in climb phase.

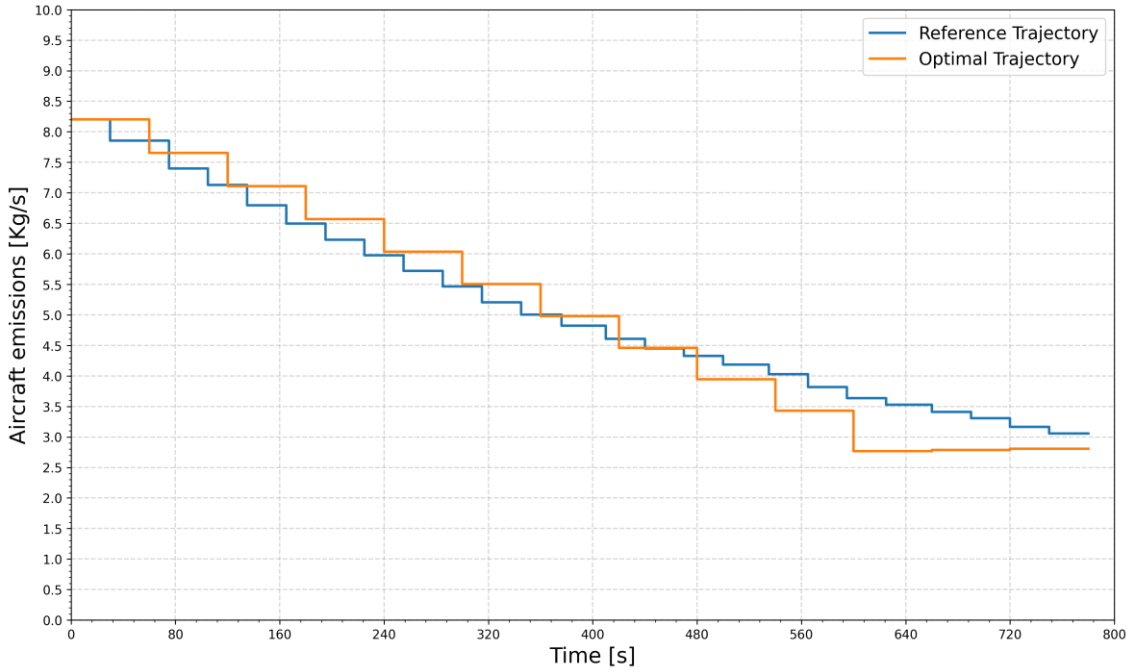


Figure 6.41: Comparison of emissions rate vs time of reference and optimal trajectory in climb phase.

Figs. (6.38) – (6.40) represent respectively the time history of controls acceleration u1 in $\left[\mathrm{m/s^2}\right]$, flight path angle rate u2 in $\left[\mathrm{rad/s}\right]$, and heading angle rate u3 in $\left[\mathrm{rad/s}\right]$ of the reference and emissions optimal trajectories. Fig (6.41) shows the emissions rate comparison between the optimal and reference flight trajectories. Although in the beginning, the aircraft emissions rate is higher in the optimal trajectory, as the aircraft climbs higher the emissions rate starts decreasing, and eventually, it gets lower than the reference trajectory, which results in lower emissions in the optimal trajectory.

Table 6.10: Aircraft emissions of reference and optimal trajectory in climb phase.

Emission Gases	**Reference Trajectory [Kg]**	**Optimal Trajectory [Kg]**	**Reductions**
Carbon dioxide (CO_2)	2870.961	2842.104	28.857 (1%)
Water vapor (H_2O)	1125.635	1114.321	11.314 (1%)
Sulfur dioxide (SO_2)	0.728	0.721	0.007 (0.96%)
Oxides of nitrogen (NO_x)	16.213	16.186	0.027 (0.16%)
Carbon monoxide (CO)	1.17	1.144	0.026 (2.2%)
Hydrocarbons (HC)	0.26	0.254	0.006 (2.3%)
Total	**4014.967**	**3974.728**	**40.239 (1%)**

The aircraft emissions of reference and optimal trajectory in the climb phase are shown in the table (6.10). The results in the table suggest that the optimal climb trajectory reduces the total aircraft emissions by 40.239 $[\mathrm{kg}]$, Which is approximately a 1% reduction than the reference commercial trajectory.

6.4.2.2 Cruise Phase

The second example is a cruise phase of a flight, the time period following the climb phase. The initial and final waypoints of the problem are also set identical to the initial and final waypoints of the cruise phase of a reference commercial flight trajectory.

The initial and final 4D waypoints (x, y, h, t) of the cruise phase of flight are shown in table (6.11).

Table 6.11: Initial and final waypoints of the trajectory in cruise phase.

Waypoint	***x* [m]**	***y* [m]**	***h* [m]**	***t* [s]**
Initial	4865492.37	-663479.172	11500	0
Final	4297741.386	658030.682	11550	6660

The comparison between the emissions optimal vs reference flight trajectory in the cruise phase of flight is presented in Figs. (6.42) – (6.49). where the orange line represents the time history of the optimal trajectory, and the blue line represents the time history of the reference commercial trajectory.

The three-dimensional (x, y, h) position of the reference and optimal trajectory in the cruise phase of flight is shown in Fig. (6.42). The cruise altitude of the emissions optimal trajectory is higher than the reference trajectory, which lowers the emissions of the aircraft.

Fig. (6.43) shows the true airspeed in $[\mathrm{m/s}]$ of both optimal and reference trajectories, the optimal trajectory chooses a lower true airspeed than the reference trajectory in order to reduce the aircraft emissions. Fig. (6.44) shows the time history of the flight path angle in $[\mathrm{rad}]$ and Fig. (6.45) shows the heading angle in $[\mathrm{rad}]$ between the reference and emissions optimal trajectory. In these figures, the true airspeed, flight path angle, and heading angle of the optimal trajectory are within the bounded constraints that were imposed on them.

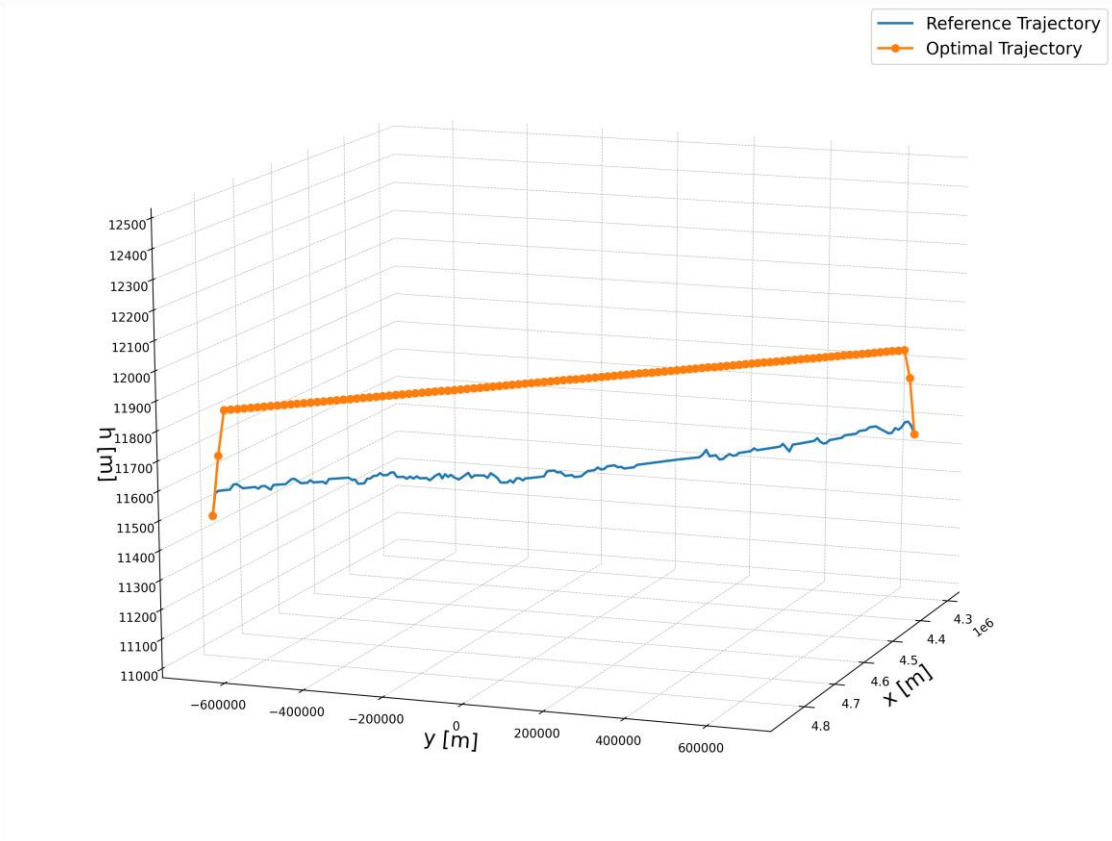


Figure 6.42: Comparison of reference and optimal 3D trajectory in cruise phase.

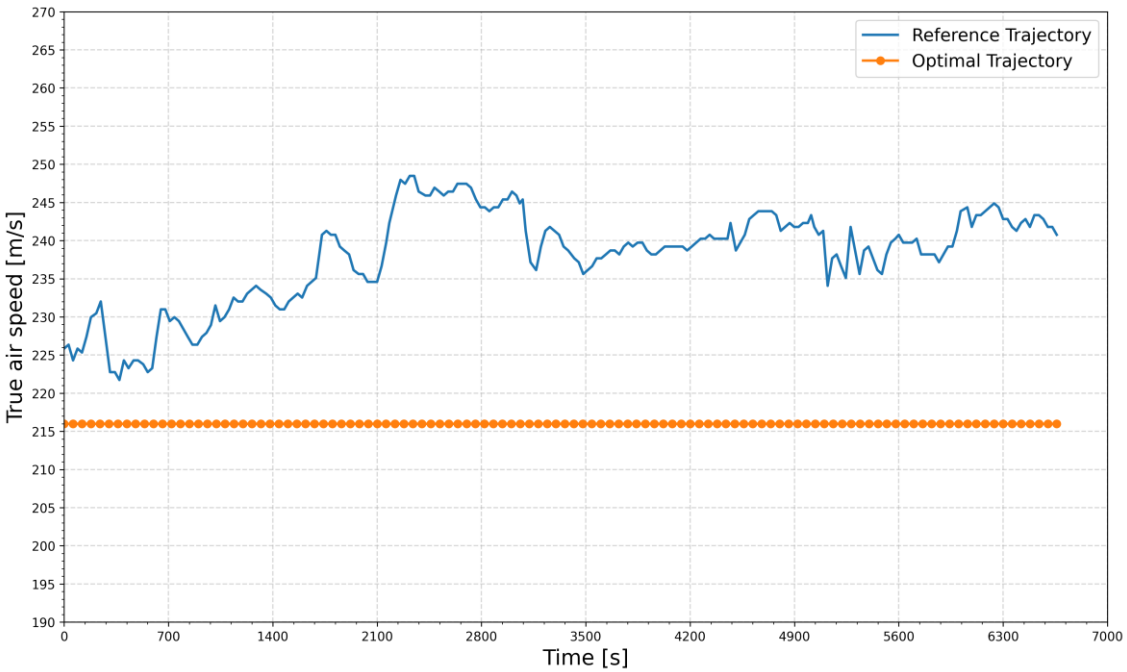


Figure 6.43: Comparison of true airspeed vs time of reference and optimal trajectory in cruise phase.

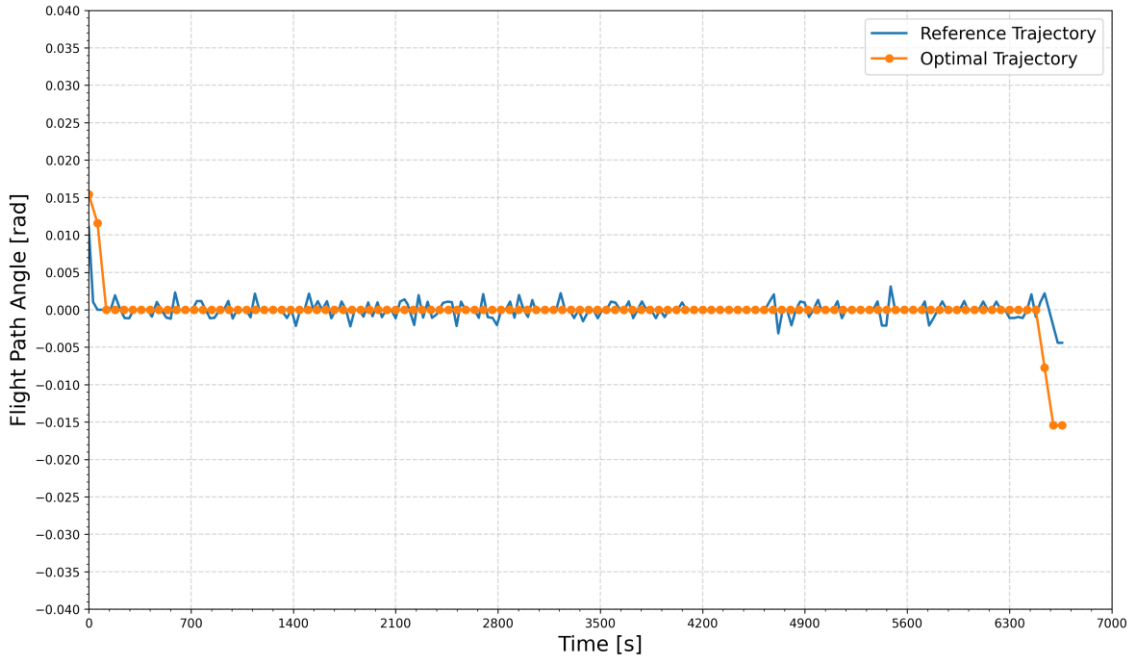


Figure 6.44: Comparison of flight path angle vs time of reference and optimal trajectory in cruise phase.

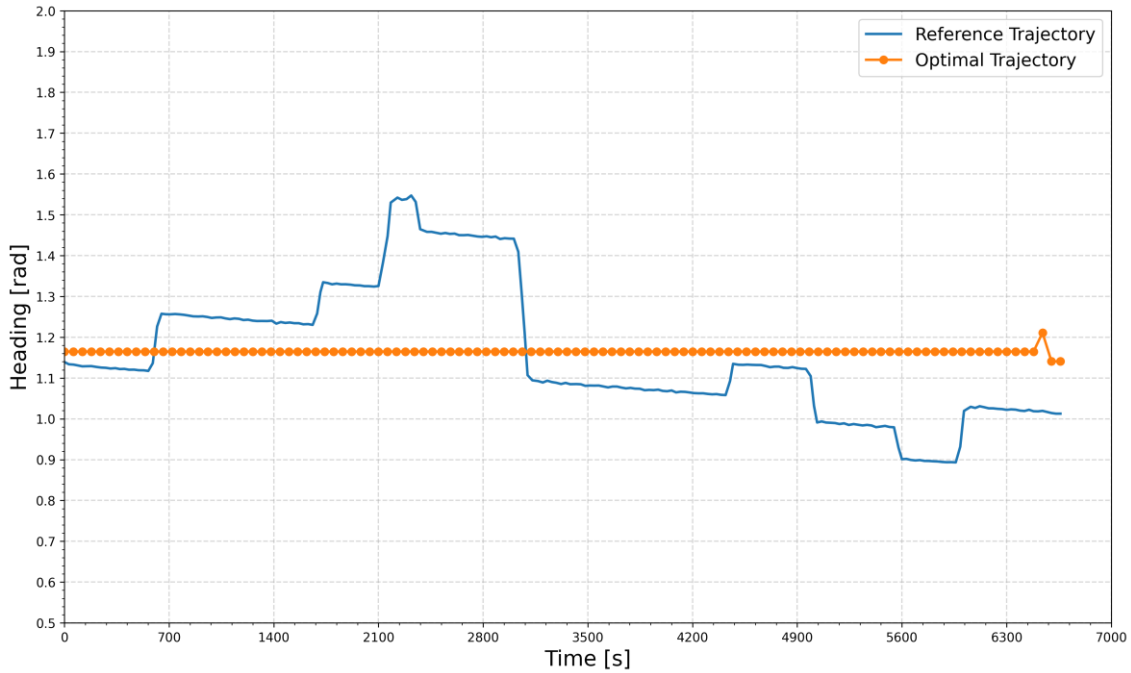


Figure 6.45: Comparison of heading vs time of reference and optimal trajectory in cruise phase.

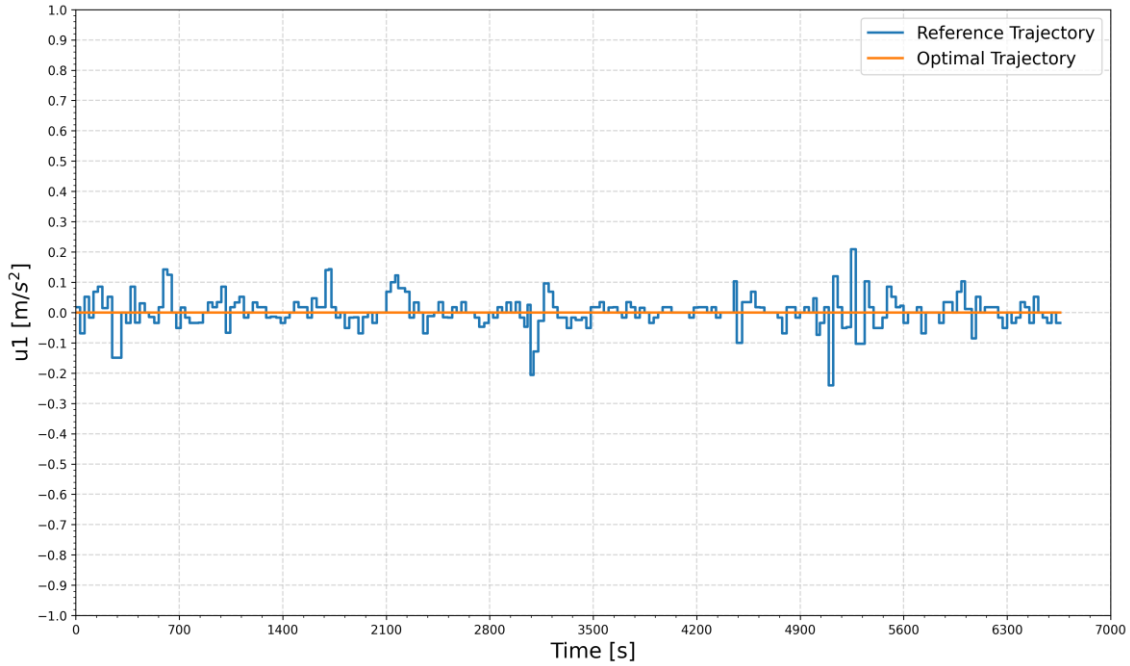


Figure 6.46: Comparison of control u1 vs time of reference and optimal trajectory in cruise phase.

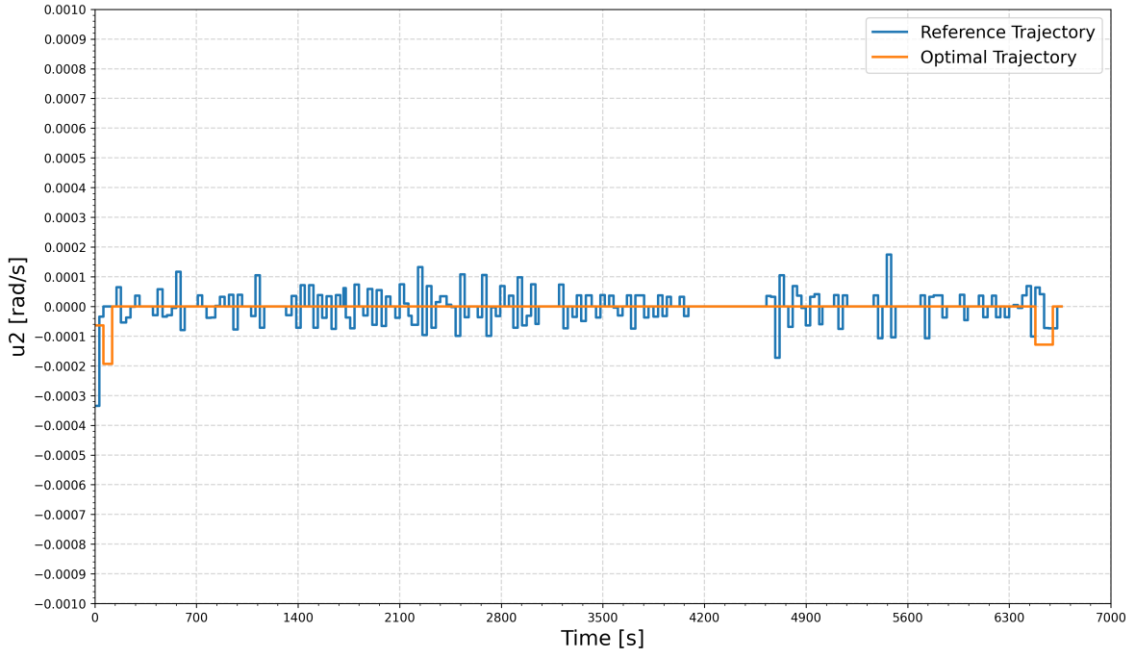


Figure 6.47: Comparison of control u2 vs time of reference and optimal trajectory in cruise phase.

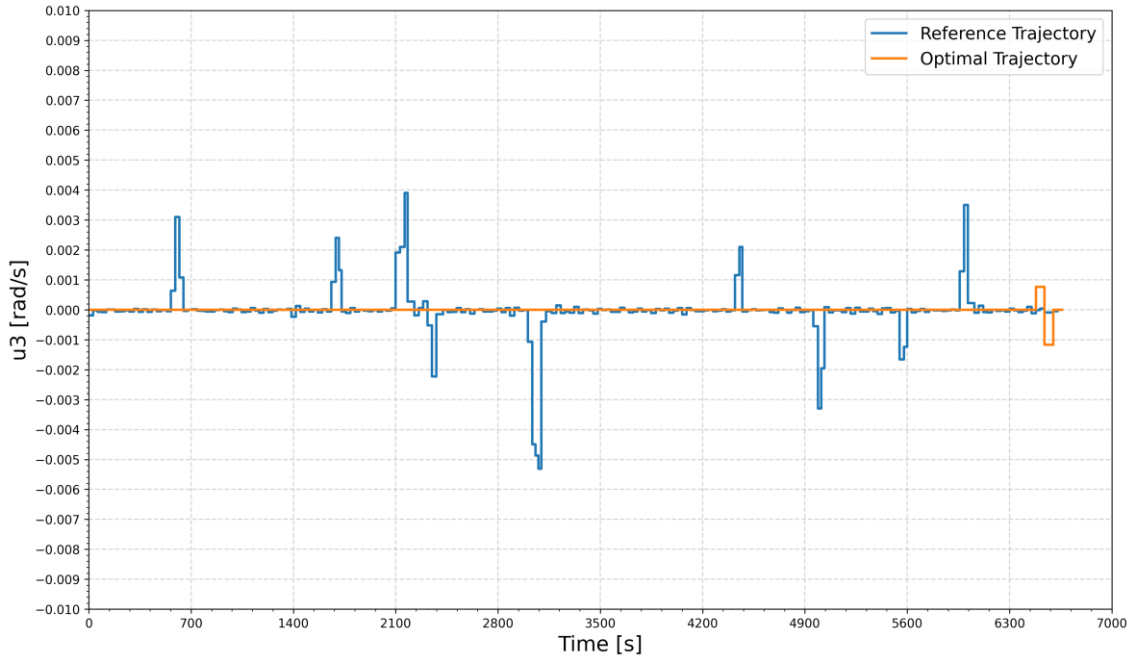


Figure 6.48: Comparison of control u3 vs time of reference and optimal trajectory in cruise phase.

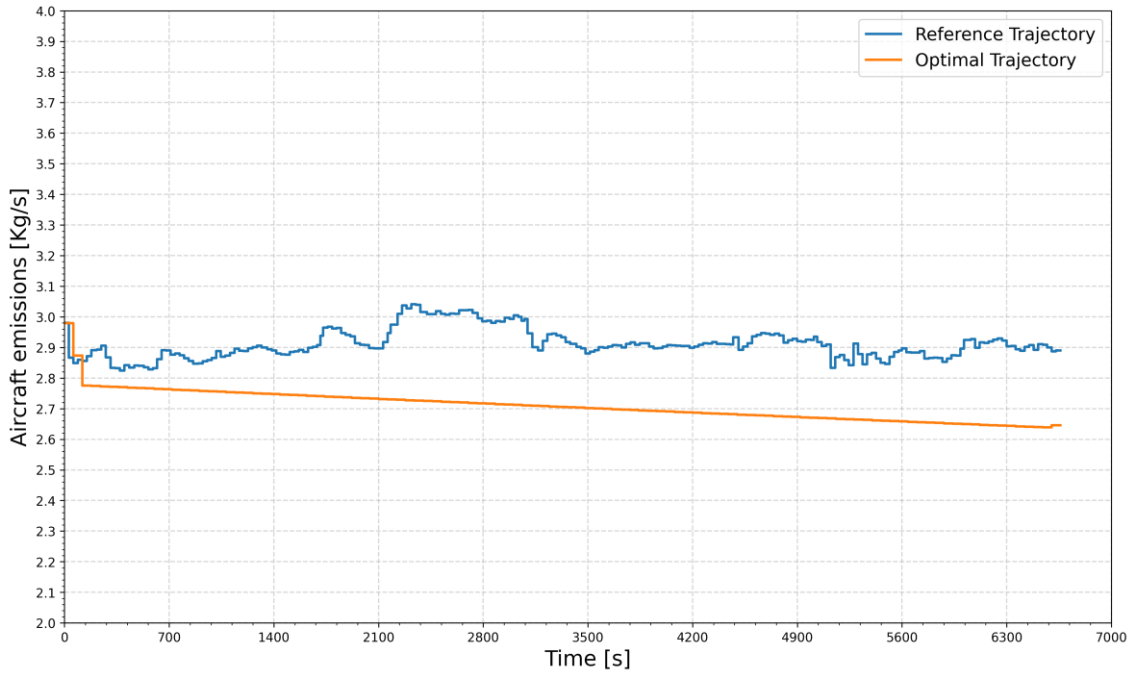


Figure 6.49: Comparison of emissions rate vs time of reference and optimal trajectory in cruise phase.

Figs. (6.46) – (6.48) represent respectively the time history of controls acceleration u1 in $\left[\mathrm{m/s^2}\right]$, flight path angle rate u2 in $\left[\mathrm{rad/s}\right]$, and heading angle rate u3 in $\left[\mathrm{rad/s}\right]$ of the reference and

emissions optimal trajectories. Fig. (6.49) shows the emissions rate comparison between the reference and emissions optimal trajectory. It can be seen from the figure that the emissions optimal trajectory chooses a cruise altitude and true airspeed that results in a lower emissions rate in most of the trajectory.

Table 6.12: Aircraft emissions of reference and optimal trajectory in cruise phase.

Emission Gases	**Reference Trajectory [Kg]**	**Optimal Trajectory [Kg]**	**Reductions**
Carbon dioxide (CO_2)	13870.029	12907.819	962.21 (6.9%)
Water vapor (H_2O)	5438.106	5060.847	377.259 (6.9%)
Sulfur dioxide (SO_2)	3.517	3.273	0.244 (6.9%)
Oxides of nitrogen (NO_x)	67.874	62.644	5.23 (7.7%)
Carbon monoxide (CO)	7.39	6.992	0.398 (5.4%)
Hydrocarbons (HC)	1.642	1.554	0.088 (5.3%)
Total	**19388.558**	**18043.129**	**1345.429 (6.9%)**

Table (6.12) shows the results of aircraft emissions for both the trajectory, the total emissions by the aircraft for the reference cruise phase trajectory is 19388.558 $[\text{kg}]$, and the emissions optimal trajectory is 18043.129 $[\text{kg}]$. This implies that by flying the optimal trajectory approximately 1345.429 $[\text{kg}]$ (6.9%) of emissions can be reduced than the reference trajectory.

6.4.2.3 Descent Phase

The third example is a descent phase of a flight, the time period following the cruise phase. Like the other cases, it is assumed that the optimal trajectory starts at the initial waypoint and ends at the final waypoint of a reference commercial descent flight trajectory. The initial and final 4D waypoints (x, y, h, t) of the descent phase are shown in table (6.13).

Table 6.13: Initial and final waypoints of the trajectory in descent phase.

Waypoint	***x* [m]**	***y* [m]**	***h* [m]**	***t* [s]**
Initial	4297741.386	658030.682	11550	0
Final	4163996.876	882046.951	2000.0	1380

The comparison between emissions optimal vs reference flight trajectory in the descent phase of flight is presented in Figs. (6.50) – (6.57). where the orange line represents the optimal trajectory, and the blue line represents the time history of the reference commercial trajectory.

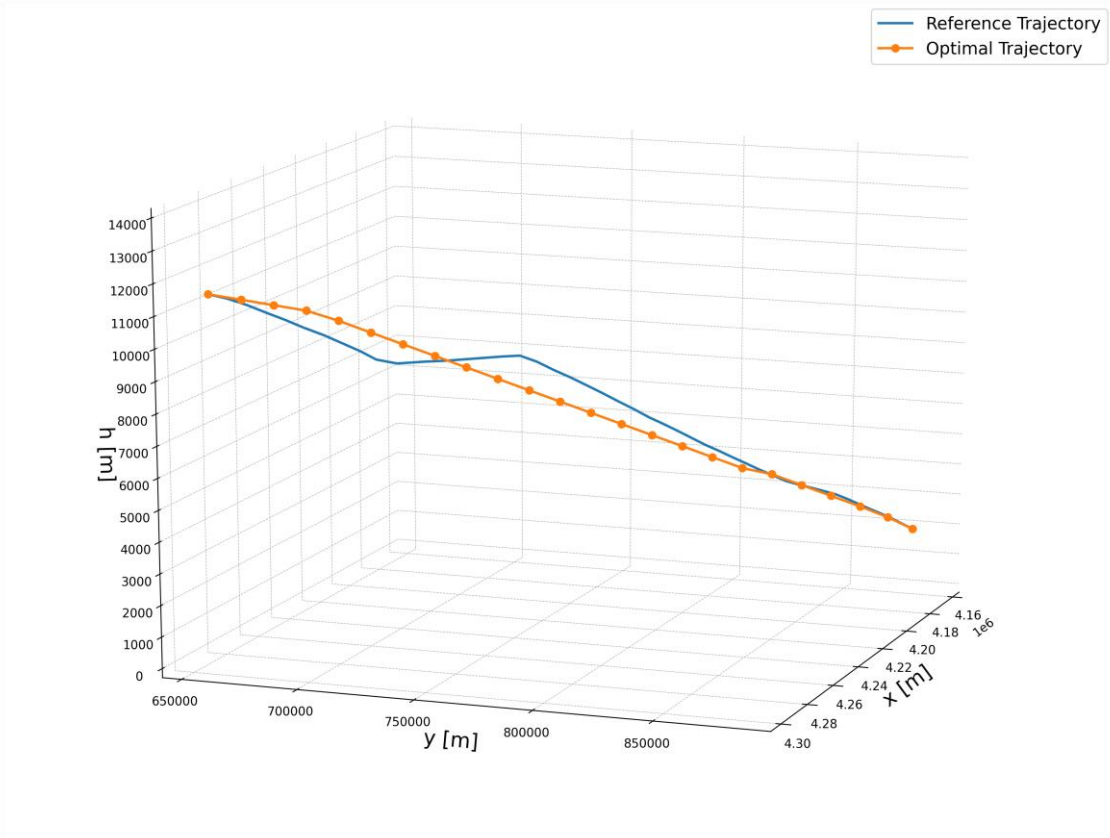


Figure 6.50: Comparison of reference and optimal 3D trajectory in descent phase.

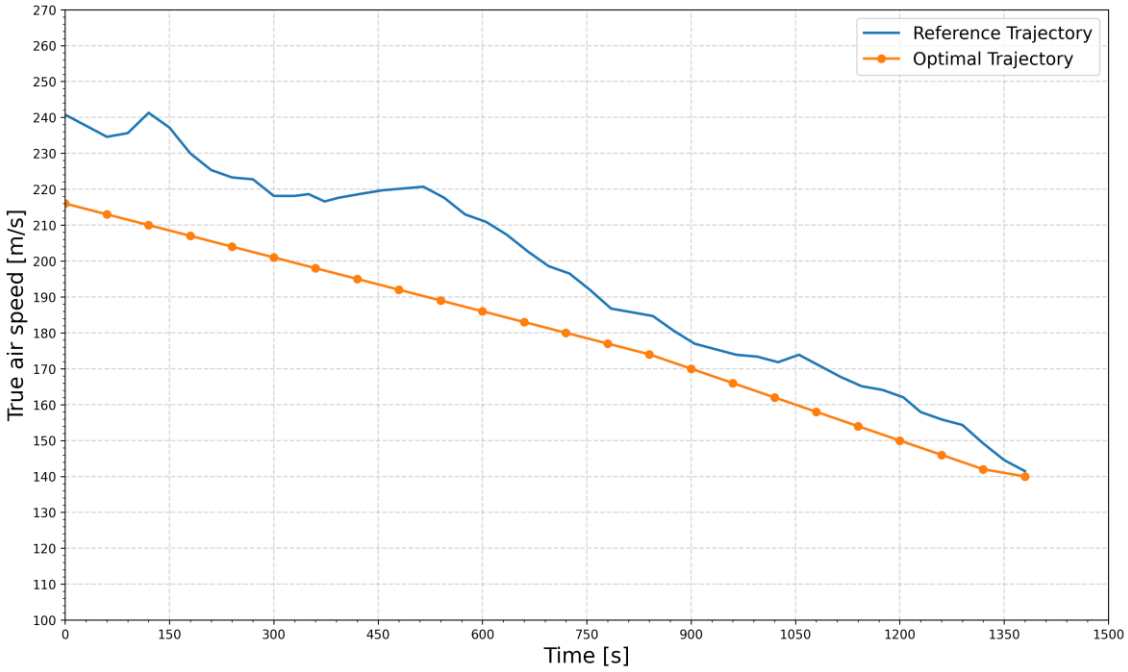


Figure 6.51: Comparison of true airspeed vs time of reference and optimal trajectory in descent phase.

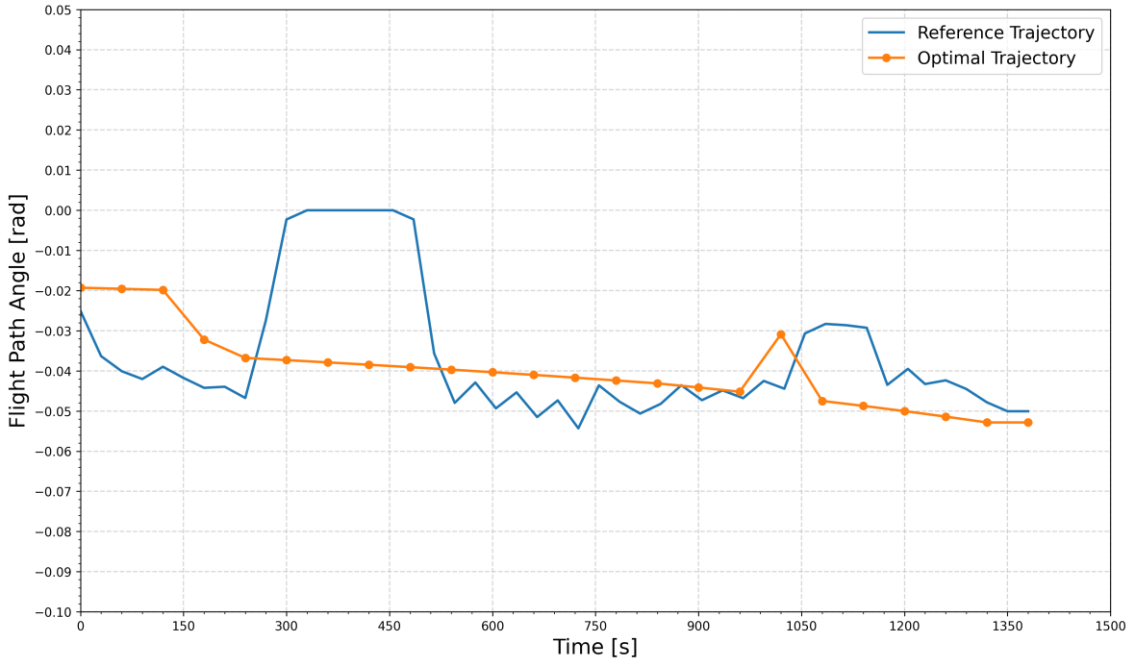


Figure 6.52: Comparison of flight path angle vs time of reference and optimal trajectory in descent phase.

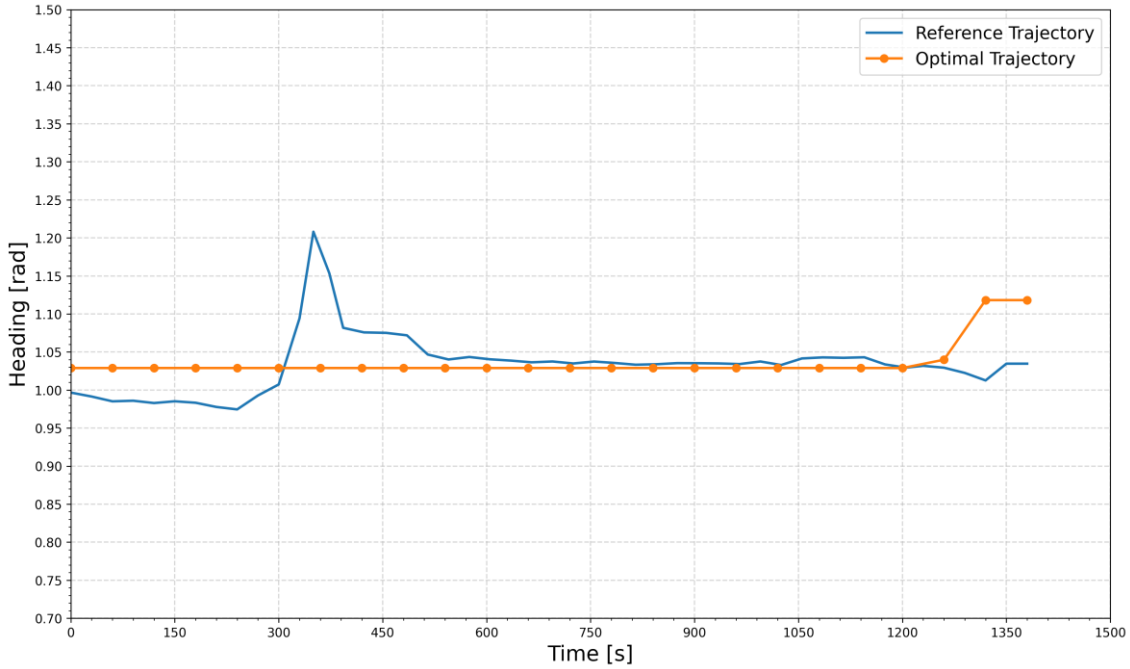


Figure 6.53: Comparison of heading vs time of reference and optimal trajectory in descent phase.

The three-dimensional (x, y, h) position of reference and emissions optimal trajectory is shown in Fig. (6.50). The figure suggests that the top of descent (ToD) occurs later in the optimal trajectory and the aircraft descend continuously, which results in fewer aircraft emissions. Fig. (6.51) shows the true airspeed in $[\mathrm{m/s}]$, Fig. (6.52) shows the time history of the flight path angle

in [rad] and Fig. (6.53) shows the heading angle in [rad] between the reference and emissions optimal trajectory. In these figures, the true airspeed, flight path angle, and heading angle of the optimal trajectory are within the bounded constraints that were imposed on them.

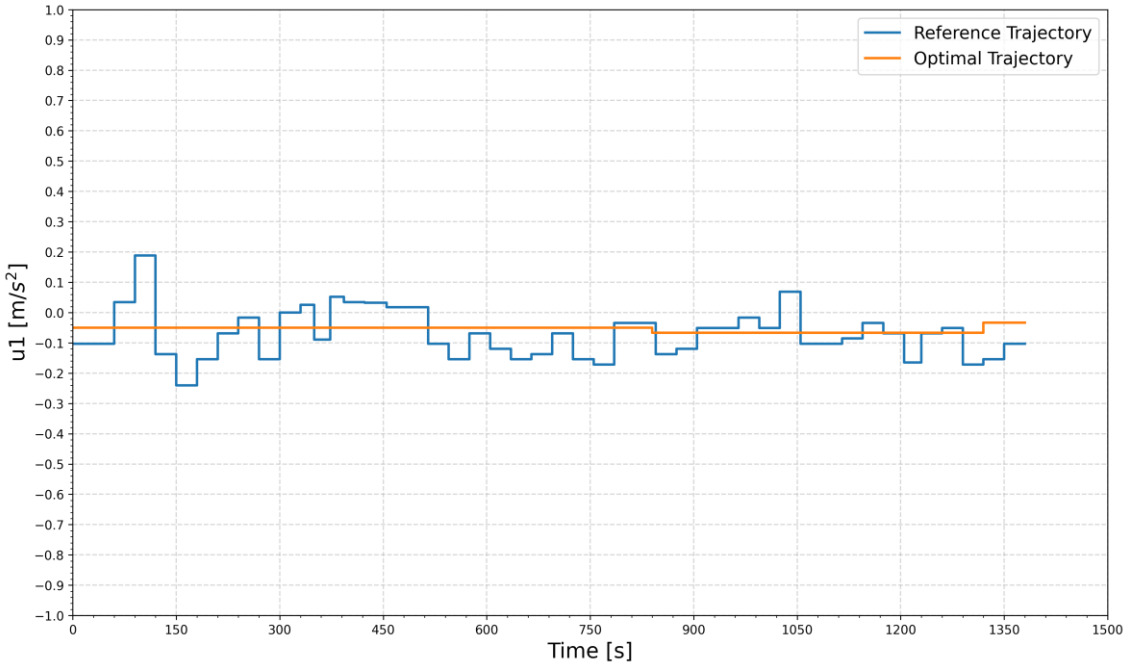


Figure 6.54: Comparison of control u1 vs time of reference and optimal trajectory in descent phase.

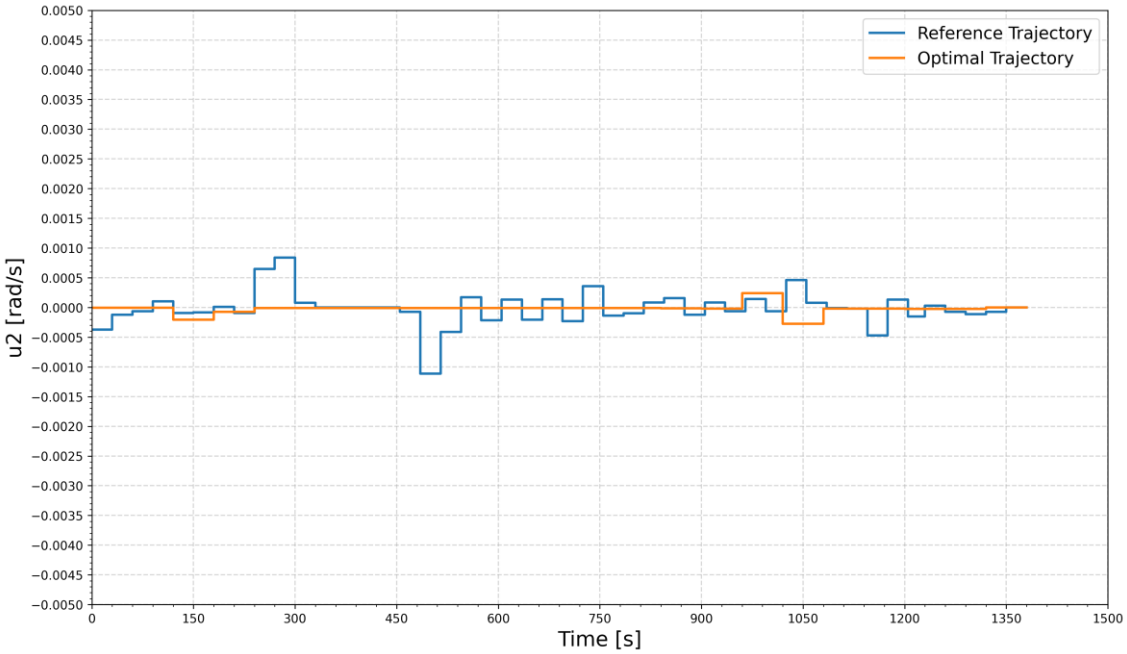


Figure 6.55: Comparison of control u2 vs time of reference and optimal trajectory in descent phase.

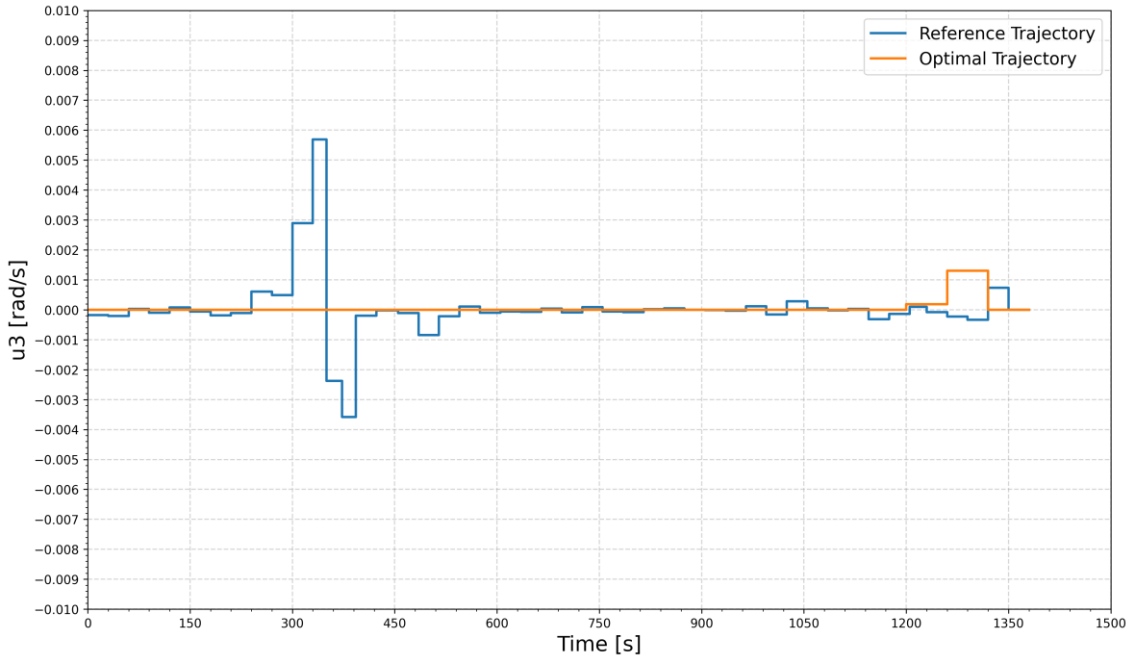


Figure 6.56: Comparison of control u3 vs time of reference and optimal trajectory in descent phase.

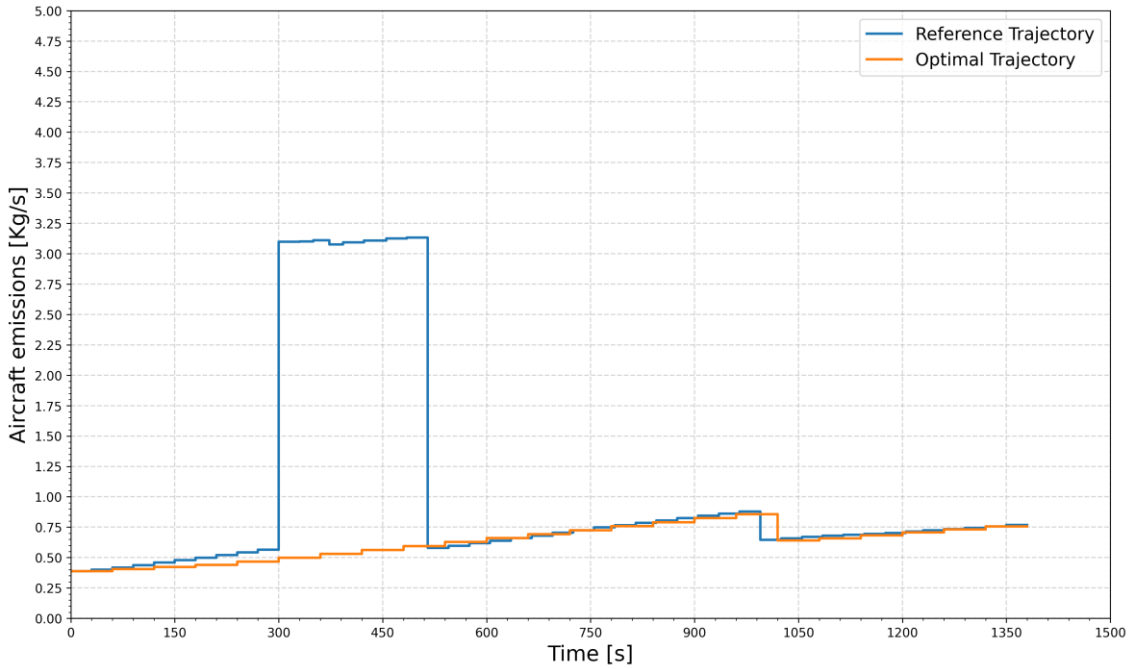


Figure 6.57: Comparison of emissions rate vs time of reference and optimal trajectory in descent phase.

Figs. (6.54) – (6.56) represent respectively the time history of controls acceleration u1 in $\left[\mathrm{m/s^2}\right]$, flight path angle rate u2 in $\left[\mathrm{rad/s}\right]$, and heading angle rate u3 in $\left[\mathrm{rad/s}\right]$ of the reference and

emissions optimal trajectories. Fig. (6.57) shows the emissions rate comparison between the reference and emissions optimal trajectory. It can be seen from the figure that the emissions rate is similar in both the optimal trajectory and the reference trajectory. However, the emissions rate of the reference trajectory spiked on several occasions due to the level off during descent.

Table 6.14: Aircraft emissions of reference and optimal trajectory in descent phase.

Emission Gases	**Reference Trajectory [Kg]**	**Optimal Trajectory [Kg]**	**Reductions**
Carbon dioxide (CO_2)	1024.359	618.24	406.119 (39.6%)
Water vapor (H_2O)	401.626	242.397	159.229 (39.6%)
Sulfur dioxide (SO_2)	0.26	0.157	0.103 (39.6%)
Oxides of nitrogen (NO_x)	5.679	3.492	2.187 (38.5%)
Carbon monoxide (CO)	0.431	0.252	0.179 (41.5%)
Hydrocarbons (HC)	0.096	0.056	0.04 (41.7%)
Total	**1432.45**	**864.593**	**567.857 (39.6%)**

Table (6.14) compares the aircraft emissions of reference and optimal trajectory in descent phase. The results suggest that by flying the optimal trajectory a 567.857 $[kg]$ reduction of aircraft emissions can be achieved, which is equivalent to 39.6% of aircraft emissions reduction than the reference descent trajectory.

6.4.2.4 Global Trajectory

This case study considered the global trajectory consists of all three climb, cruise, and descent phases of flight. The initial and final waypoints of the problem are set identical to the initial and final waypoints of the reference commercial flight trajectory. The initial and final 4D waypoints (x, y, h, t) are shown in table (6.15).

Table 6.15: Initial and final waypoints of the global trajectory.

Waypoint	***x* [m]**	***y* [m]**	***h* [m]**	***t* [s]**
Initial	4912599.596	-781419.581	2100.0	0
Final	4163996.876	882046.951	2000.0	8820

Figs. (6.58) – (6.65) show the comparison between emissions optimal vs reference commercial trajectory, where the orange line represents the time history of the optimal trajectory, and the

blue line represents the time history of reference commercial trajectory as exported from the website Flightaware.

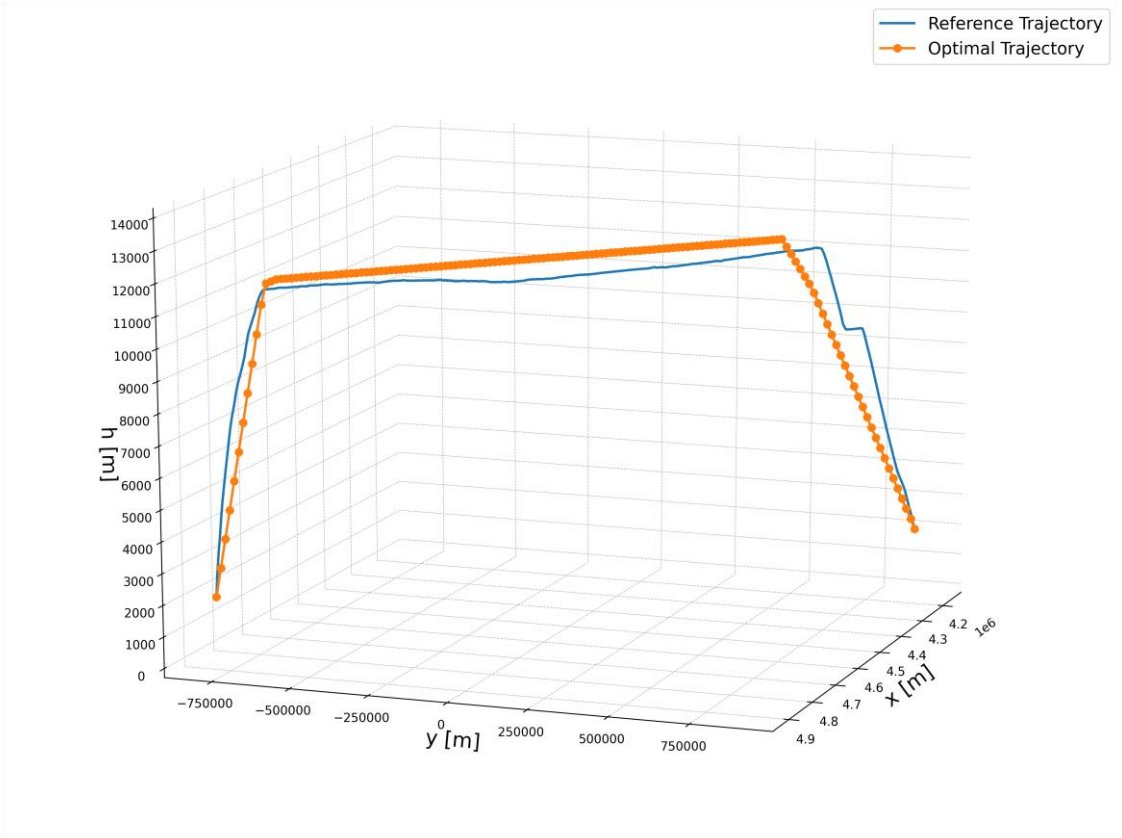


Figure 6.58: Comparison of reference and optimal global 3D trajectory.

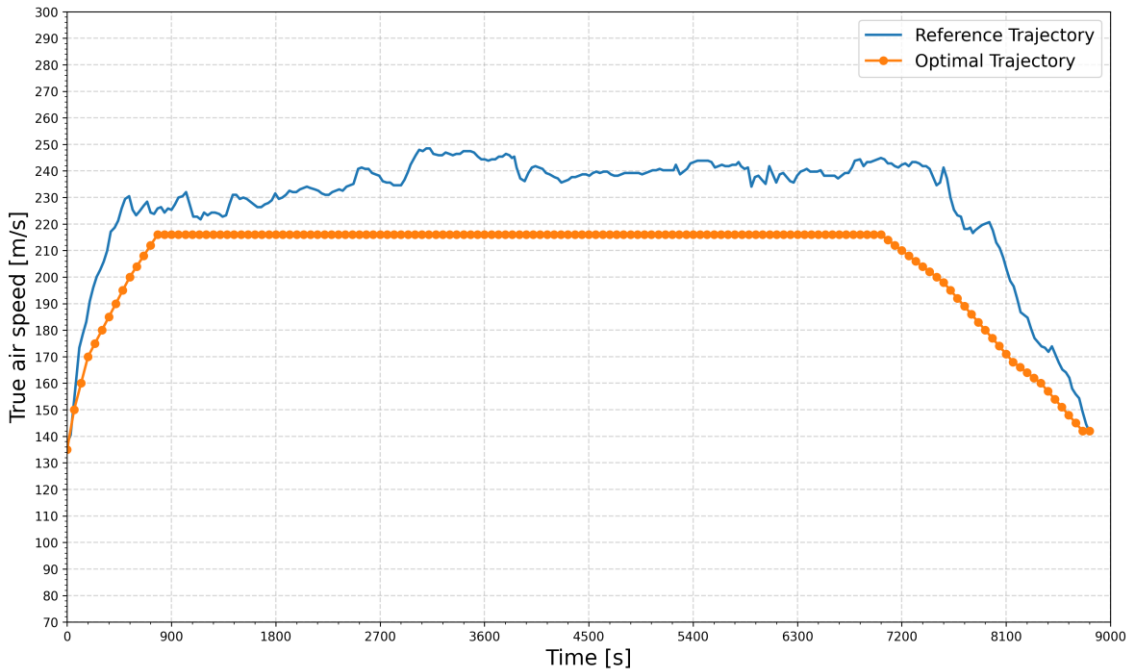


Figure 6.59: Comparison of true airspeed vs time of reference and optimal global trajectory.

Fig. (6.58) shows the three-dimensional (x, y, h) position of the reference and optimal trajectory. Where the aircraft reaches the top of climb (ToC) at 780 $[s]$ and the top of descent (ToD) at 7020 $[s]$ in the optimal trajectory. The cruise altitude of the optimal trajectory is at 11850 $[m]$ with a constant speed of 216 $[m/s]$.

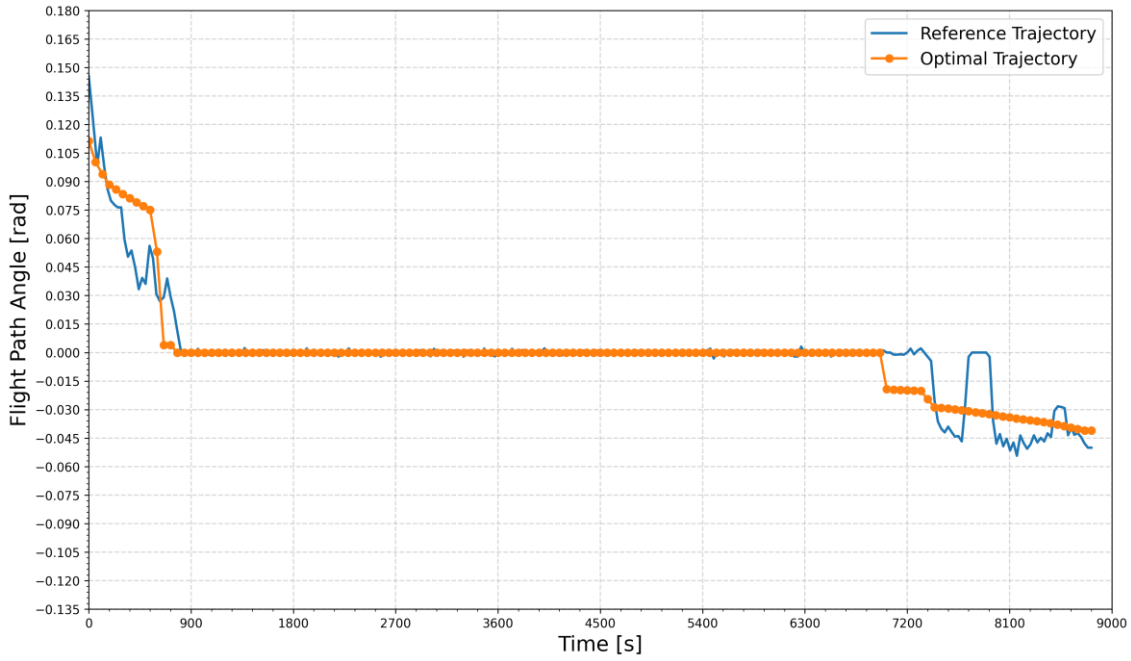


Figure 6.60: Comparison of flight path angle vs time of reference and optimal global trajectory.

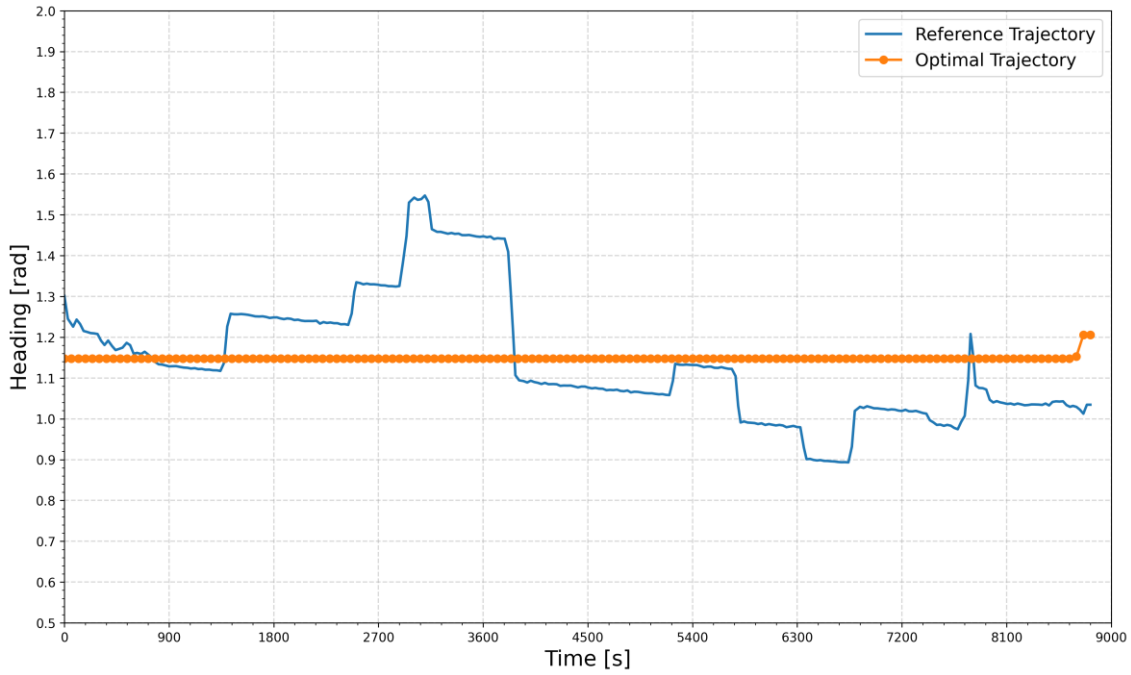


Figure 6.61: Comparison of Heading vs time of reference and optimal global trajectory.

Fig. (6.59) shows the true airspeed in $[\mathrm{m/s}]$, Fig. (6.60) shows the time history of the flight path angle in $[\mathrm{rad}]$and Fig. (6.61) shows the heading angle in $[\mathrm{rad}]$ between the reference and emissions optimal trajectory. In these figures, the true airspeed, flight path angle, and heading angle of the optimal trajectory are within the bounded constraints that were imposed on them.

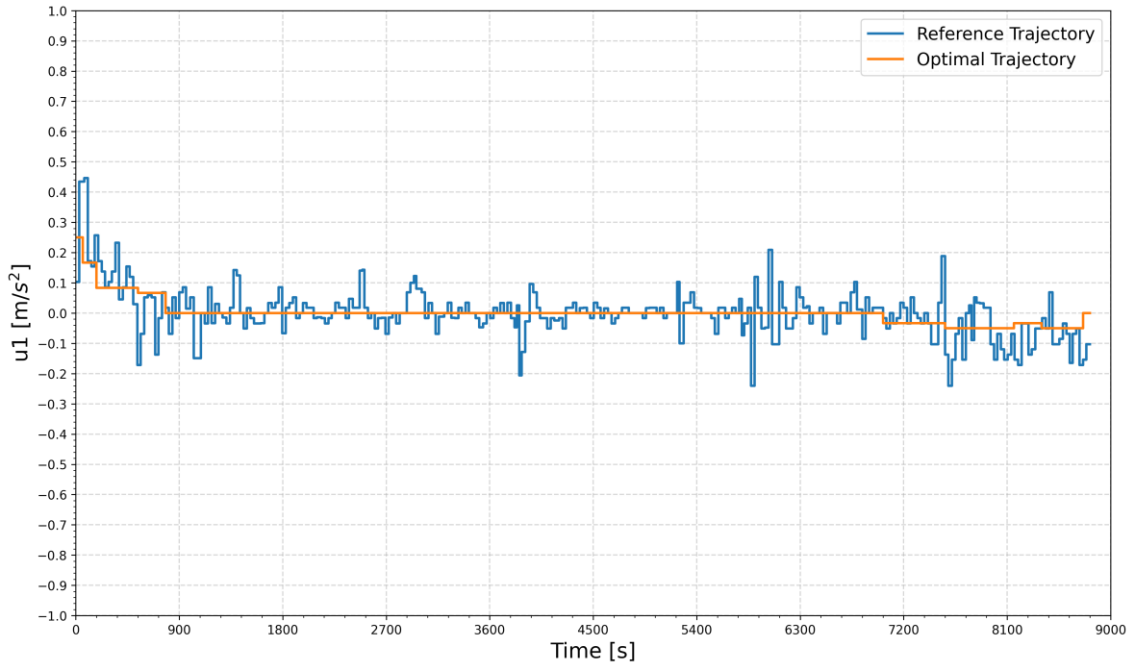


Figure 6.62: Comparison of control u1 vs time of reference and optimal global trajectory.

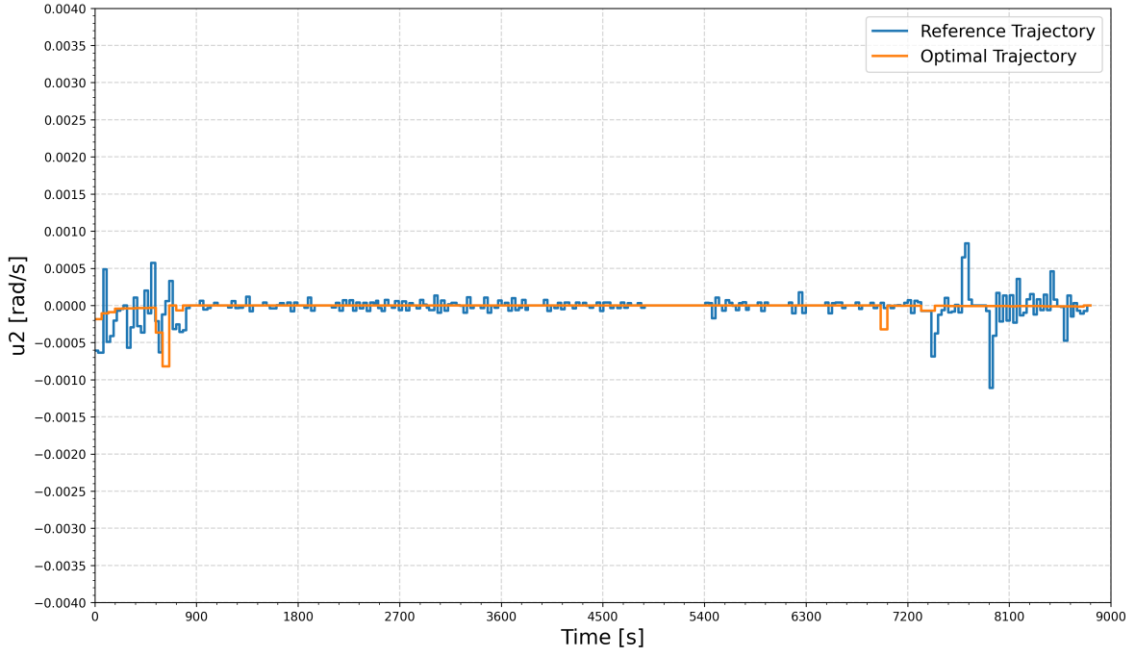


Figure 6.63: Comparison of control u2 vs time of reference and optimal global trajectory.

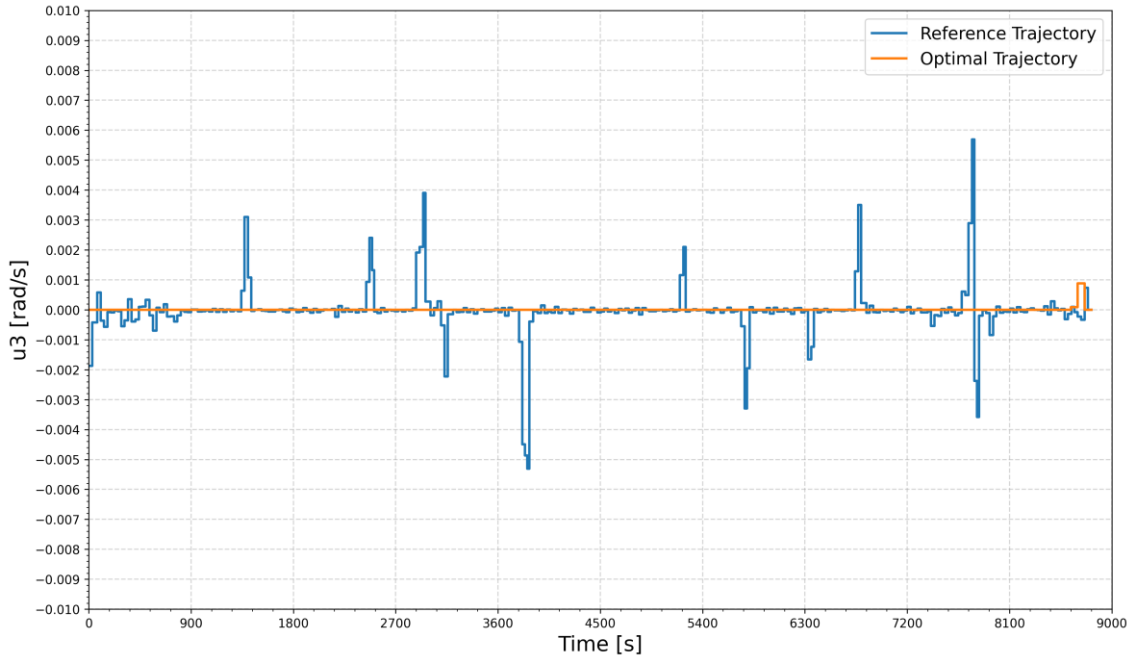


Figure 6.64: Comparison of control u3 vs time of reference and optimal global trajectory.

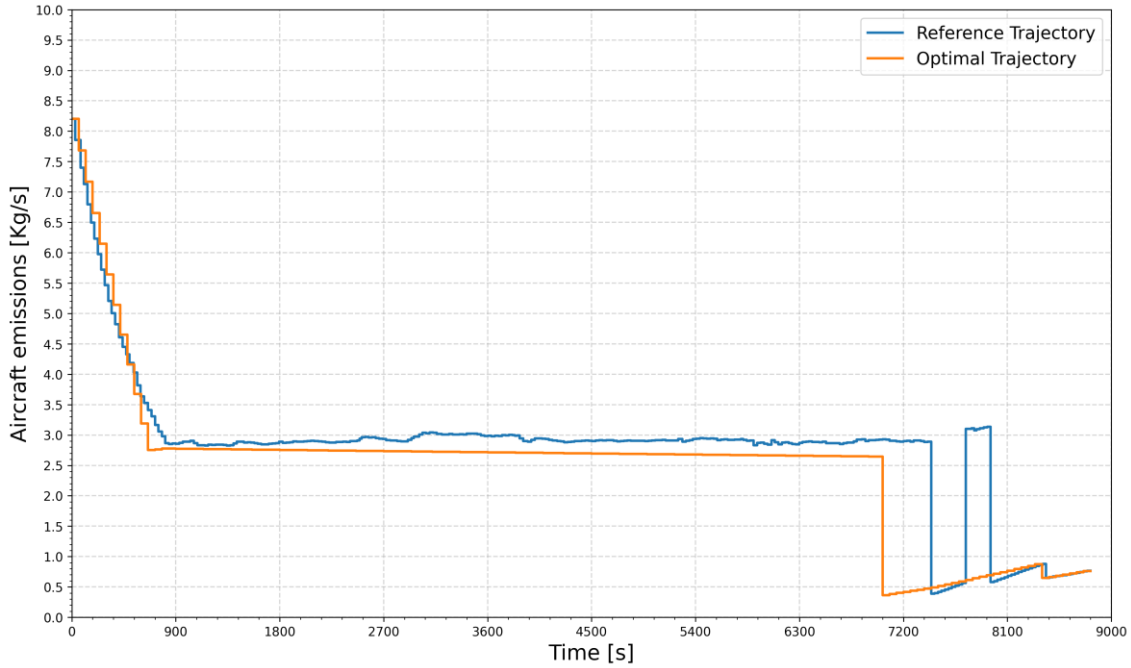


Figure 6.65: Comparison of emissions rate vs time of reference and optimal global trajectory.

Figs. (6.62) – (6.64) represent respectively the time history of controls acceleration u1 in $\left[\mathrm{m/s^2}\right]$, flight path angle rate u2 in $\left[\mathrm{rad/s}\right]$, and heading angle rate u3 in $\left[\mathrm{rad/s}\right]$ of the reference and

emissions optimal trajectories. Fig (6.65) shows the aircraft emissions rate comparison between the trajectories. It can be seen from the figure that in most of the trajectory the emissions rate is lower in the optimal trajectory than the reference trajectory, which results in reduced total aircraft emissions.

Table 6.16: Aircraft emissions of reference and optimal global trajectory.

Emission Gases	**Reference Trajectory [Kg]**	**Optimal Trajectory [Kg]**	**Reductions**
Carbon dioxide (CO_2)	17765.477	15812.284	1953.193 (11%)
Water vapor (H_2O)	6965.418	6199.618	765.8 (11%)
Sulfur dioxide (SO_2)	4.505	4.009	0.496 (11%)
Oxides of nitrogen (NO_x)	89.767	79.857	9.91 (11%)
Carbon monoxide (CO)	8.99	8.048	0.942 (10.5%)
Hydrocarbons (HC)	1.998	1.788	0.21 (10.5%)
Total	**24836.155**	**22105.605**	**2730.55 (11%)**

The aircraft emissions of the reference and optimal trajectory are presented in table (6.16). Based on the results from the table, the aircraft emissions are 2730.551 $[kg]$ less in optimal trajectory, which is approximately a 10.99% reduction of aircraft emissions by flying the emissions optimal trajectory.

6.5 Summary

In this chapter, a Modified Dynamic Programming (MDP) approach has been proposed in order to cope with the two serious drawbacks of traditional Dynamic Programming (DP). The MDP reduces the first drawback of the curse of dimensionality by limiting the search space at each stage and considering only the grid points of that reduced search space. The second drawback of the menace of the expending grid is also solved by the MDP approach by generating the control values inside the allowable range. This generation of the control values guarantees reachable grid points for the states at any stage.

The proposed MDP approach has been applied to a couple of case studies to validate its applicability. In the first case study, the MDP is applied to generate optimal trajectories that minimize the aircraft fuel consumption in four examples (climb phase, cruise phase, descent phase, and global trajectory). In all examples, the generated fuel optimal trajectories were compared to the corresponding reference airliner trajectories. The results suggest that the optimal trajectories improve the flight efficiency by reducing fuel consumption by 3.6% for the climb

phase, 5.3% for the cruise phase, and 26.9% for the descent phase. In the fourth example, while considering the global trajectory the optimal trajectory reduces the total fuel consumption by 10.1% than the reference airliner flight trajectory.

In the second case study, the MDP approach is applied to generate aircraft emissions optimal trajectory. The results suggest that the optimal trajectories reduce the aircraft emissions in all four examples than the corresponding reference airliner trajectories by the margin of 1% for the climb phase, 6.9% for the cruise phase, 39.6% for the descent phase, and 10.99% for the global trajectory.

This new modified approach of DP has the potential to be one of the core components of a future autonomous air transportation system, as the numerical example demonstrates it can successfully generate fuel optimal trajectory with little computational effort, which implies it can also be applied to on-line trajectory generation. The results of this study are published in a journal [141].

Chapter 7

7. Fuel Flow Rate Prediction Using Radial Basis Function Neural Network.

7.1 Introduction

This chapter deals with Artificial Neural Network (ANN) to predict the fuel flow rate of a commercial aircraft in different phases of flight. An ANN is a collection of connected computational units or nodes called neurons arranged in multiple computational layers, it is designed to simulate the way the human brain analyzes and processes information. Fig. (7.1) illustrates the biological neuron that is composed of the nucleus, dendrites, synapses, and axons. Dendrites are the receptor; they receive signals from many other neurons and passes the signals to the nucleus. The nucleus processes the received signals and generates a response signal. Then the signals will be transmitted through the axon to the dendrites to become the next neuron input signal.

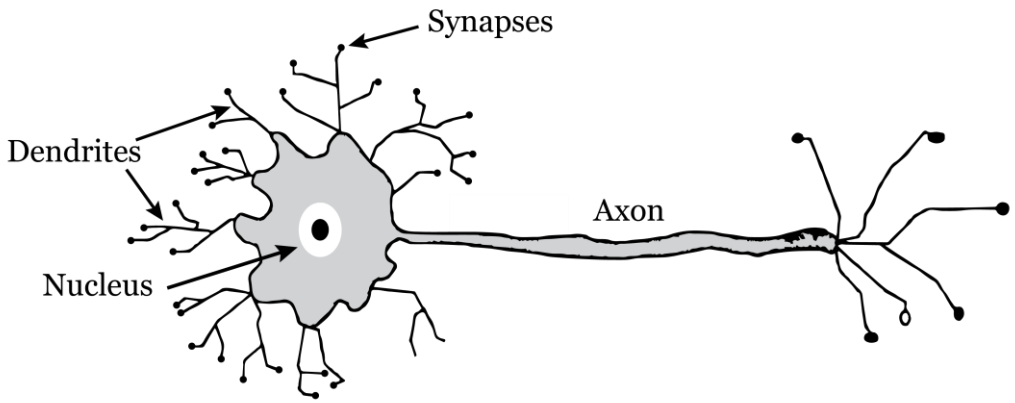


Figure 7.1: Structure of biological neuron.

Fig. (7.2) illustrates the mathematical model of an artificial single neuron, which is a computational model inspired by biological neurons. The artificial neuron receives one or more inputs and sums them and adds a bias term to this sum to produce an output. Usually, each input is separately weighted, and the sum is passed through a non-linear function known as an activation function δ . Activation functions determine whether a node should be activated, they may improve or reduce the performance of the artificial neuron. Their main task is to transform the signal in the node into an output signal. Then it will be used in the next layer of the network or will be output from it. There are different types of activation functions (e.g., step function,

sigmoid, tanh, relu, etc.). For the given single artificial neuron as shown in Fig. (7.2), there is a total of n inputs $(x_1, x_2, ..., x_n)$, n weights $(w_1, w_2, ..., w_n)$, and a bias term w_0.

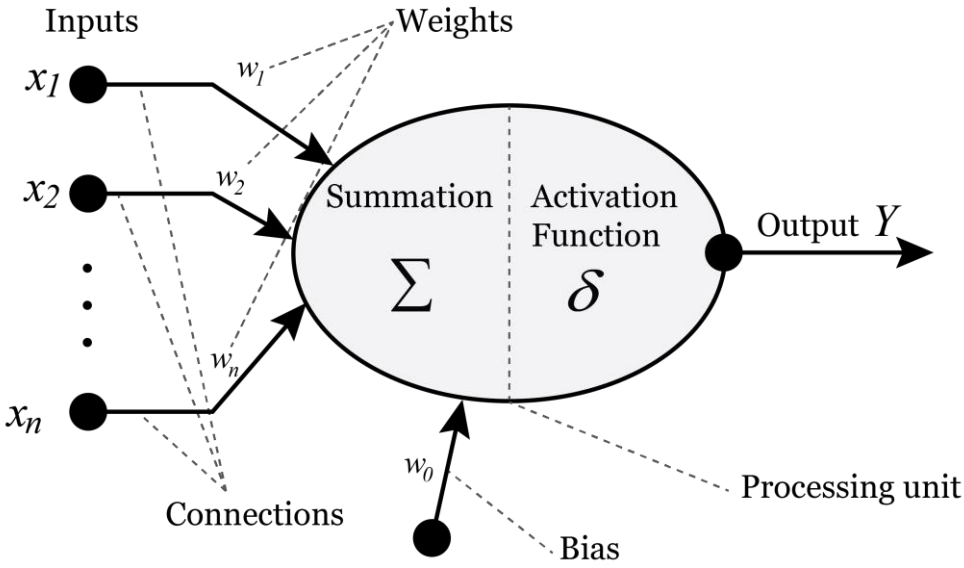


Figure 7.2: Structure of artificial neuron.

ANNs are composed of interconnected single artificial neurons. Artificial neurons are usually organized in multiple layers and connect only to neurons that are directly in the preceding and next layer. Each layer contains a specific number of single neurons. The neuron passes the signal at the neuron on the right and in this way, information is propagated by the network. Without using the activation function δ, the ANN would become a linear function with many variables [142], [143], [144]. An example of a single hidden layer ANN is illustrated in Fig. (7.3).

In this study, Radial Basis Function (RBF) neural network [145], [146], [147], [148] was applied to predict the fuel flow rate of a commercial aircraft for the three main (e.g., climb, cruise, and descent) phases of flight. The RBF neural network is composed of only three layers with just a single hidden layer in the middle and the structure is fixed only by determining the hidden layer and the weights between the hidden and the output layers. The RBF neural network can approximate any regular function and converge rapidly, it also has the advantage of automatically determine the number of neurons in the hidden layer and can avoid complicated calculations. Thus, the time required for training can be much faster than other ANN. The RBF neural network is the most suitable for a complex structure because of the adaptive learning capability and can approximate the non-linear system. When actual flight data or manual data from an aircraft are accessible, this RBF neural network could effectively predict the fuel flow rate in actual

applications. This study takes the true airspeed and altitude as input parameters and provides the fuel flow rate as an output parameter for the cruise, climb, and descent phases.

The next section presents the RBF neural network. In the later section, the RBF neural network was applied to predict a commercial aircraft fuel flow rate of the climb, cruise, and descent phases of flight. Later the predicted fuel flow rate was used to generate an aircraft optimal 4D green trajectory that optimizes the green cost of a commercial flight trajectory between Lisbon and Munich.

7.2 Radial Basis Function Neural Network

Radial Basis Function (RBF) neural network [145], [146], [147], [148] is a commonly used type of Artificial Neural Network (ANN) for function approximation problems. RBF neural networks are distinguished from other neural networks due to their universal approximation and faster learning speed. This faster learning speed is due to fact that RBF neural network just has one hidden layer.

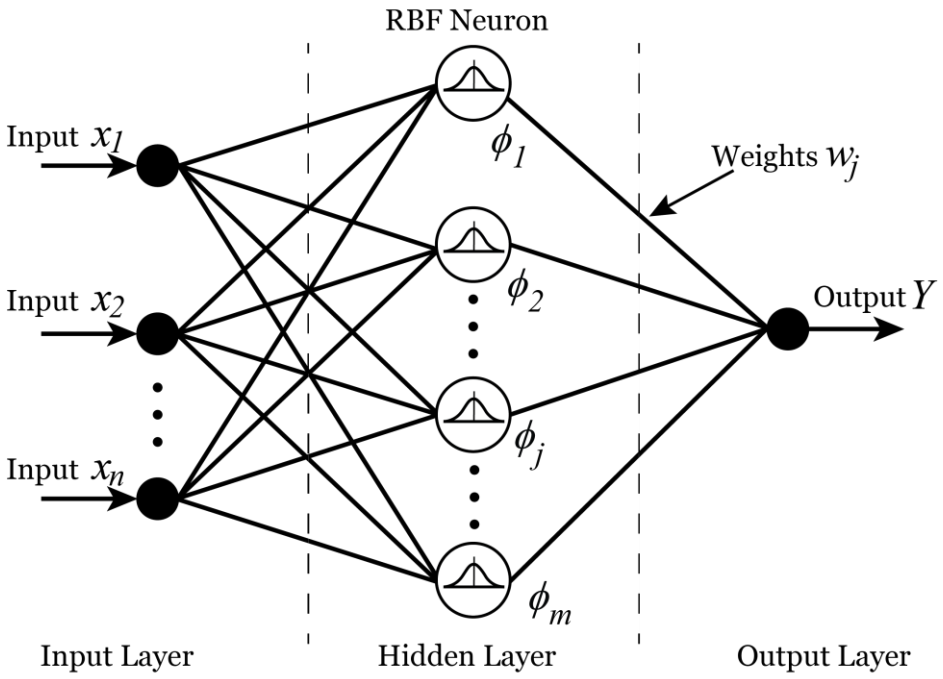


Figure 7.3: Structure of radial basis function neural network.

An RBF neural network is a type of Feed-Forward Neural Network (FFNN) composed of three different layers: an input layer, a hidden layer, and an output layer. In a FFNN, signals are transmitted only in one direction, that is from the input nodes through the hidden nodes, and then to output nodes, there are no feedback connections in which outputs of the model are fed back into itself [149]. Fig. (7.3) illustrates the structure of RBF neural network design with multiple inputs and a single output.

In the RBF neural network, the input layer consists of an input vector $X = [x_1, x_2, ..., x_n]$. The hidden layer is composed of m RBF neurons. Each input is connected to the hidden layer of RBF neurons. The hidden layer of the RBF network has several forms of radial basis activation functions such as multiquadric, inverse multiquadric, and Gaussian function, etc. Among them, Gaussian functions are generally the most used in RBF neural network hidden layer as illustrated in Fig. (7.4).

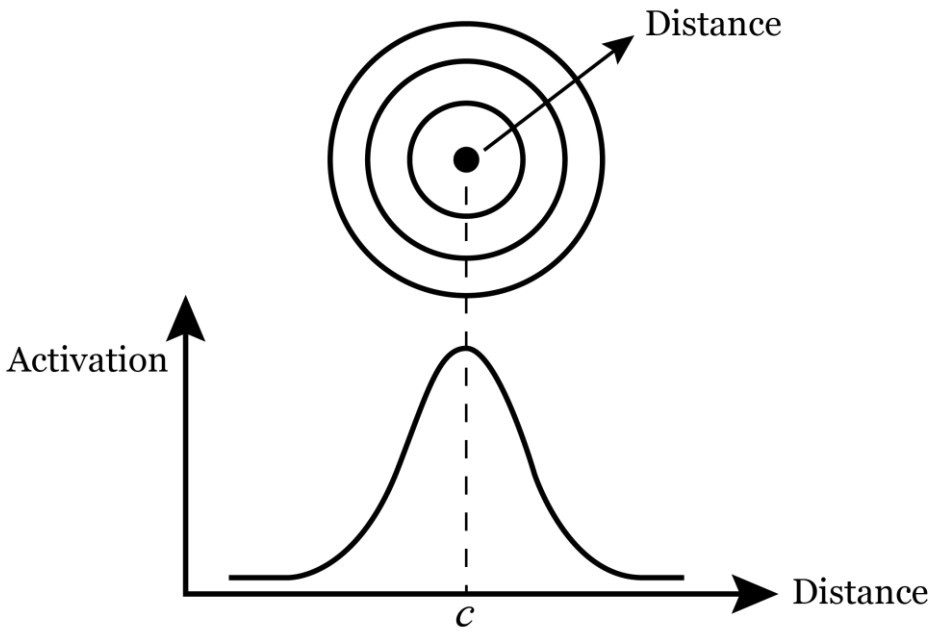


Figure 7.4: Gaussian radial basis function.

The Gaussian radial basis function ϕ_j can be defined as follow:

$$\phi_j = \exp\left(-\frac{r_j^2}{2\sigma_j^2}\right) \tag{7.1}$$

Where $\sigma_j > 0$ is the width of the j^{th} neuron in the hidden layer and r_j is the radial distance calculated by the Euclidean norm of the distance between the input vector and the neuron center as follow:

$$r_j = \|X - c_j\| \tag{7.2}$$

Where X is the input. c_j is the center of the j^{th} neuron in the hidden layer $j = (1, 2, ..., m)$. At the location c_j, the value of the RBF ϕ_j is maximum.

The output of the neural network Y is shaped by a linearly weighted sum of the number of basis function in the hidden layer as follow:

$$Y = \sum_{j=1}^{m} w_j \ \phi_j \tag{7.3}$$

Where w_j is the weight of the j^{th} hidden layer neuron to the output, and RBF ϕ_j is the output of the j^{th} node in the hidden layer given in Eq. (7.1). With the described structure, the hidden layer consists of m hidden nodes, which use nonlinear transformations to the input space. However, the output of the network is a linear combination of the radial basis functions computed by the hidden layer nodes.

The training procedure of the RBF neural network involves determining the following parameters.

- Determining the number of neurons m in the hidden layer. This is an important factor affecting the predictive properties of the network. The neural network uses many RBF neurons to estimate the best predictive performance. RBF can rapidly converge and automatically determine the number of neurons in the hidden layer.

- Determining the center c_j and width σ_j of each RBF neuron in each node in the hidden layer. In the training procedure, a key point is determining the positions of the node centers c_j of kernel functions in an RBF network, because the outputs of the kernel functions supply essential information to the network's generalization capabilities. There are many methods to find the centers and width of the RBF neurons in the hidden layer, and the *k-means* clustering is the most used algorithm.

- Determining the weight values w_j applied to the RBF outputs as they are passed to the output layer. The Least Mean Square (LMS) is widely used to optimize the weights between the hidden and the output layer nodes.

The accuracy of the model was evaluated based on the coefficient of determination R^2 and Relative Approximation Error (RAE). Coefficient of determination R^2 is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by the independent variable or variables in a regression model and expressed as follow:

$$R^2 = \frac{\sum_{k=1}^{N}\left[Y\left(X_k\right) - \bar{Y}\right]^2}{\sum_{k=1}^{N}\left[Y_{actual,k} - \bar{Y}\right]^2} \tag{7.4}$$

When $R^2 > 0.8$ the model accuracy is considered excellent. Relative Approximation Error (RAE) is the root mean square error divided by the mean of target values.

$$RAE = \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[Y_{actual,k} - Y\left(X_k\right)\right]^2}}{\overline{Y}} \tag{7.5}$$

Where $Y_{actual,k}$ are the actual target values of the network output of the input vector X_k, $Y\left(X_k\right)$ is the predicted values of the input vector X_k, $\overline{Y}$ is the mean of $Y_{actual,k}$ values, and N is the number of training samples.

When 100% of the data set is considered to calculate the RAE Eq. (7.5) is called global RAE_G. The RAE considering 80% of the data set called the training RAE_T, and the RAE considering the rest 20% of the data set called the test RAE_{test}. When $0 \le RAE_G \le 0.1$, $0 \le RAE_T \le 0.1$, and $0 \le RAE_{test} \le 0.2$ the model accuracy is considered excellent.

7.3 Fuel Flow Rate Prediction

In this section, the Radial Basis Function (RBF) neural network is applied to predict the aircraft fuel flow rate in the climb, cruise, and descent phases of flight. A twinjet subsonic aircraft was considered (as in subsection 6.4.2) to analyze the problem. The true airspeed and flight altitude were taken as input parameters in the RBF neural network and the network output the fuel flow rate.

Table (7.1) presents the Fuel flow rate prediction performance using the RBF neural network containing a total of 248, 156, and 248 training samples in the climb, cruise, and descent phases, respectively.

Table 7.1: Performance of fuel flow rate prediction using RBF neural network.

	Climb Phase	**Cruise Phase**	**Descent Phase**
Training samples N	248	156	248
Coefficient of determination R^2	0.9999999999999994	0.99999999999939	0.99999999999980
RAE_G	7.0705028536 e-09	6.4688856212 e-08	9.7090979893 e-08
RAE_T	8.3832694394 e-09	7.1982320899 e-08	1.0253309737 e-07
RAE_{test}	1.1675859013 e-08	1.6886460014 e-08	9.5146709234 e-08

The results in table (7.1) suggest that the coefficients of determination R^2 are extremely close to 1 and relative approximation errors RAE_G, RAE_T, and RAE_{test} are almost zero for all three phases of flight. Thus, it can be concluded that the obtained fuel flow rate models by the RBF neural network are efficient for obtaining accurate new values for the fuel flow rate.

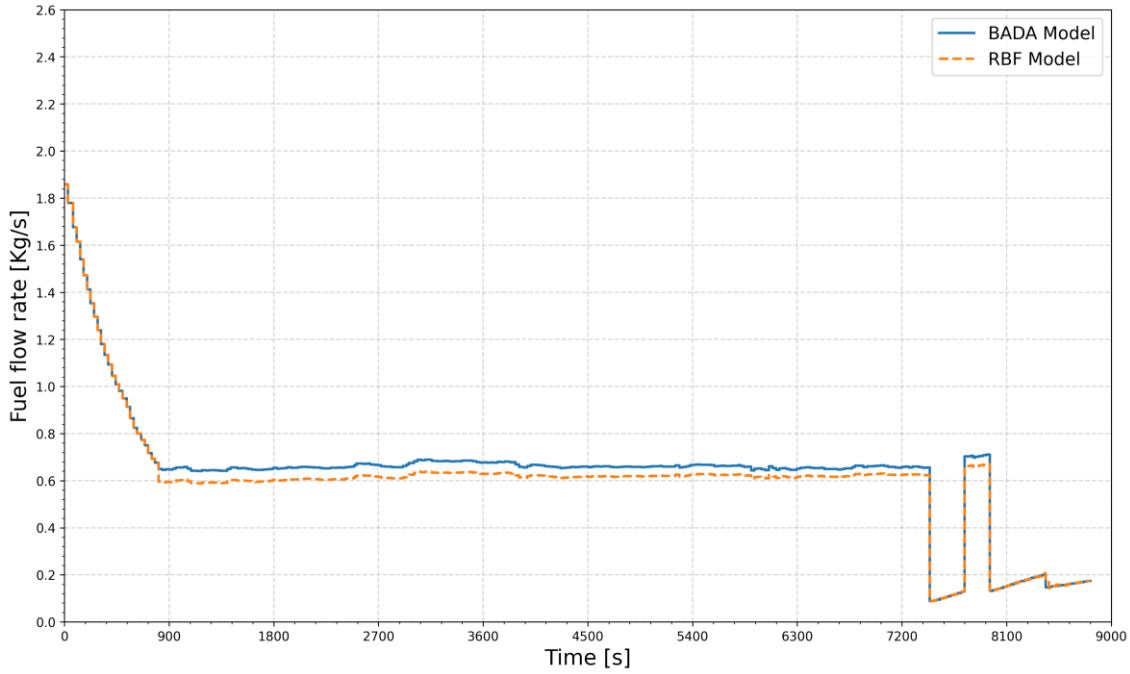


Figure 7.5: Comparison of the fuel flow rate of RBF neural network vs BADA model.

An actual commercial flight trajectory (from Lisbon to Munich) fuel flow rates are presented in Fig. (7.5). In the figure, the solid blue line and the orange dotted line represent the fuel flow rates obtained by the BADA model and RBF neural network model, respectively. Considering the result shown in Fig. (7.5), it can be concluded that there is a good fit between the actual values and those predicted by the model for the fuel flow rate in each flight phase.

7.4 4D Trajectory Optimization Using RBF Neural Network Predicted Fuel Model

This section solves a 4D trajectory optimization problem by Modified Dynamic Programming (MDP) while using the fuel flow rate model predicted by the Radial Basis Function (RBF) neural network.

Generally, the aircraft system dynamics is modeled by a set of nonlinear equations of motion (EOMs). In this chapter, a simplified version of the three degrees of freedom (3DOF) EOMs are considered Eqs. (6.14) – (6.19), where the state vector is represented by the position, velocity, flight path angle, and heading of the flight vehicle.

In Eqs. (6.14) – (6.19), the (x, y, h) is the position of the aircraft, the $V, \gamma,$ and ψ are the velocity, flight path angle, and heading respectively, the variables $u_1, u_2,$ and u_3 are respectively the acceleration, the flight path angle rate, and the heading rate. The state vector is composed by $X = [x, y, h, V, \gamma, \psi]$ and the control vector is composed by $U = [u_1, u_2, u_3]$. The real-world flight

operates under several constraints, due to aerodynamic, structural, and propulsive limitations, bound constraints are imposed on the state and control vectors Eqs. (6.20) – (6.23):

In this study, the performance index is considered to optimize the green cost GC of the aircraft, which can be defined by the following equation.

$$J_{GC} = \int_{t_0}^{t_f} GC(t)\ dt \tag{7.6}$$

Where the GC green cost can be defined as follow:

$$GC = Cost_{FF} + Cost_{CO_2} + Cost_{emi\_gas} \tag{7.7}$$

Where, $Cost_{FF}$ is the fuel cost, $Cost_{CO_2}$ is the cost of carbon dioxide emissions, and $Cost_{emi\_gas}$ is the cost of other emissions gases. These costs can be defined as follow:

$$Cost_{FF} = C_{FF}\ FF \tag{7.8}$$

$$Cost_{CO_2} = C_{CO_2}\ E_{CO_2} \tag{7.9}$$

$$Cost_{emi\_gas} = C_{gas}\ \left(E_{H_2O} + E_{SO_2} + E_{NO_x} + E_{CO} + E_{HC}\right) \tag{7.10}$$

Where the fuel cost coefficient is $C_{FF} = 0.8[€/\text{kg}]$, The carbon cost coefficient is $C_{CO_2} = 0.023[€/\text{kg}]$, and the other emission gases cost coefficient is $C_{gas} = 0.004[€/\text{kg}]$. FF is the fuel flow rate predicted by the RBF neural network. $E_{CO_2}, E_{H_2O}, E_{SO_2}, E_{NO_x}, E_{CO}$, and E_{HC} are respectively the emission of carbon dioxide, water vapor, sulfur dioxide, oxides of nitrogen, carbon monoxide, and hydrocarbons.

7.4.1 Numerical Example

This section presents a numerical example of 4D trajectory optimization. The fuel flow rate of the example is predicted using the Radial-basis function (RBF) neural network and the trajectory optimization problem is solved using the Modified Dynamic Programming (MDP). In the example, the green optimal trajectory was compared with a reference commercial flight trajectory for the same route. The flight information was taken from the website Flightaware.

The aircraft emissions model was exported from the Boeing Fuel Flow Method 2 (BFFM2). The BFFM2 uses the International Civil Aviation Organization (ICAO) emission databank to determine the reference emission indices, which eventually allow the calculation of the aircraft emissions [67]. Since the same model is used to calculate the aircraft emissions of both commercial flight trajectory and proposed optimal trajectory, the model error does not directly

affect the difference of emissions. All the analysis of the simulation has been done using Python 3.7.

In this example, a flight from Lisbon to Munich and a twinjet aircraft was considered as in subsection (6.4.2). The constraints of the case study have been selected according to the reference commercial flight trajectory for the same route as follow:

Altitude [m]	:	$2000 \leq h(t) \leq 11850$
True airspeed [m/s]	:	$135 \leq V(t) \leq 250$
Flight Path Angle [rad]	:	$-0.06 \leq \gamma(t) \leq 0.15$
Heading Angle [rad]	:	$0.8 \leq \psi(t) \leq 1.6$

In this case study, the interval between the states grid points (x, y, h) are $\Delta x = 1000$ [m], $\Delta y = 1000$ [m], $\Delta h = 50$ [m], and the interval between the stages is $\Delta t = 60$ [s].

This example considered the global trajectory consists of all three climb, cruise, and descent phases of flight. The initial and final waypoints of the problem are set identical to the initial and final waypoints of the reference commercial flight trajectory. The initial and final 4D waypoints (x, y, h, t) are shown in table (6.15) in the previous chapter.

Figs. (7.6) – (7.12) show the comparison between green optimal vs reference commercial global trajectory, where the orange line represents the time history of the optimal trajectory, and the blue line represents the time history of reference commercial trajectory as exported from the website Flightaware.

Fig. (7.6) shows the three-dimensional (x, y, h) position of the reference and optimal trajectory. Where the aircraft reaches the top of climb (ToC) at 780 [s] and the top of descent (ToD) at 7020 [s] in the optimal trajectory. The cruise altitude of the optimal trajectory is at 11850 [m] with a constant speed of 216 [m/s].

Fig. (7.7) shows the true airspeed in [m/s], Fig. (7.8) shows the time history of the flight path angle in [rad] and Fig. (7.9) shows the heading angle in [rad] between the reference and green optimal trajectory. In these figures, the true airspeed, flight path angle, and heading angle of the optimal trajectory are within the bounded constraints that were imposed on them.

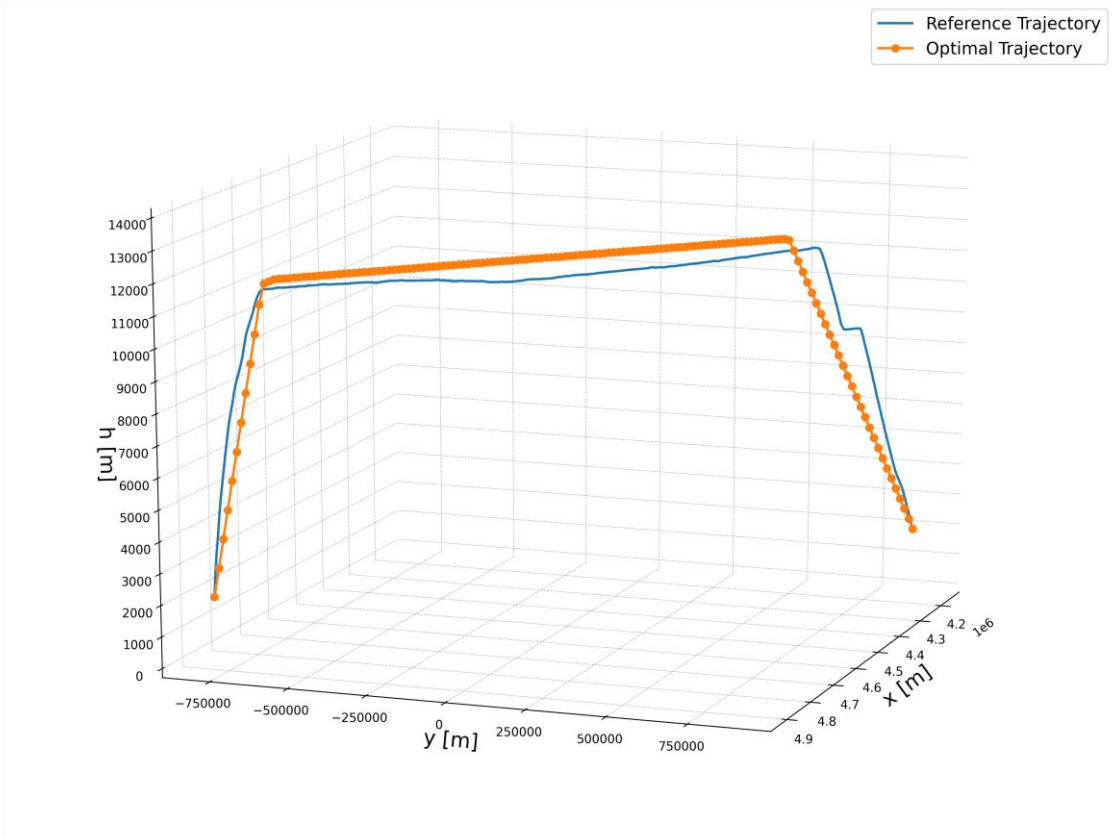


Figure 7.6: Comparison of reference and green optimal 3D trajectory.

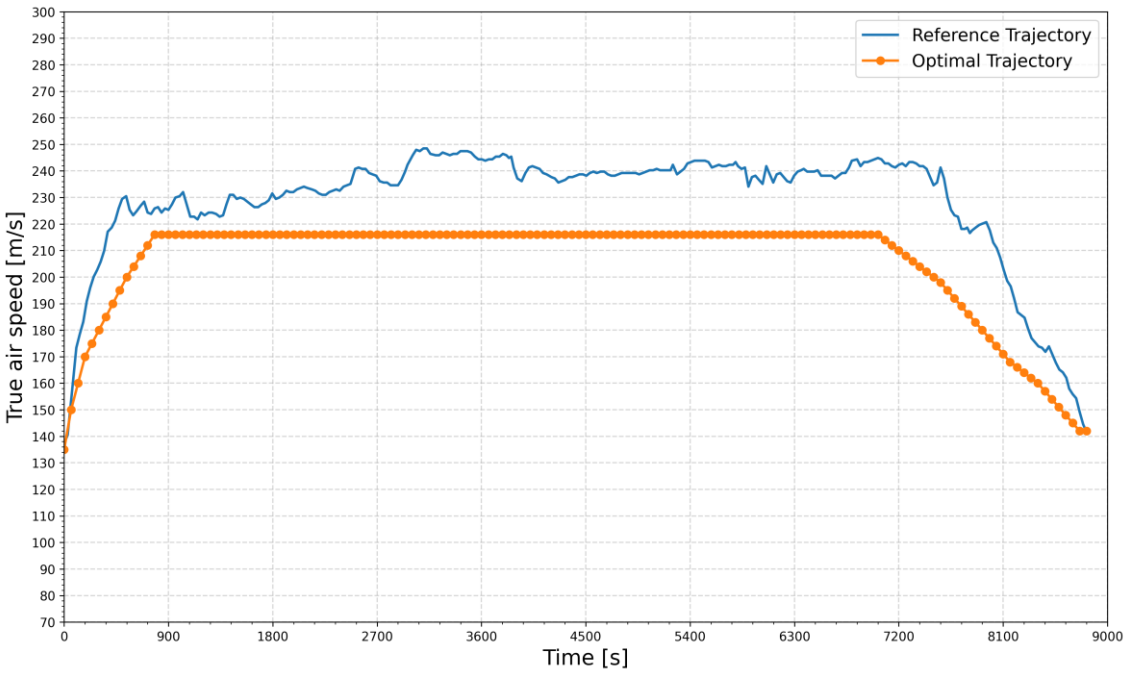


Figure 7.7: Comparison of true airspeed vs time of reference and green optimal trajectory.

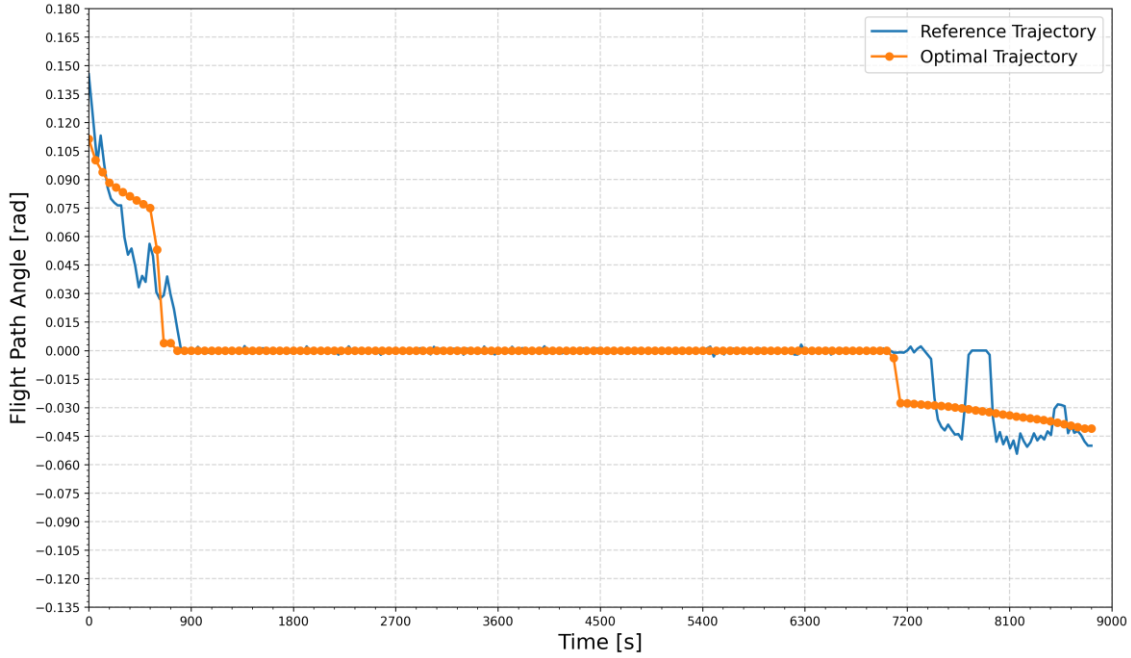


Figure 7.8: Comparison of flight path angle vs time of reference and green optimal trajectory.

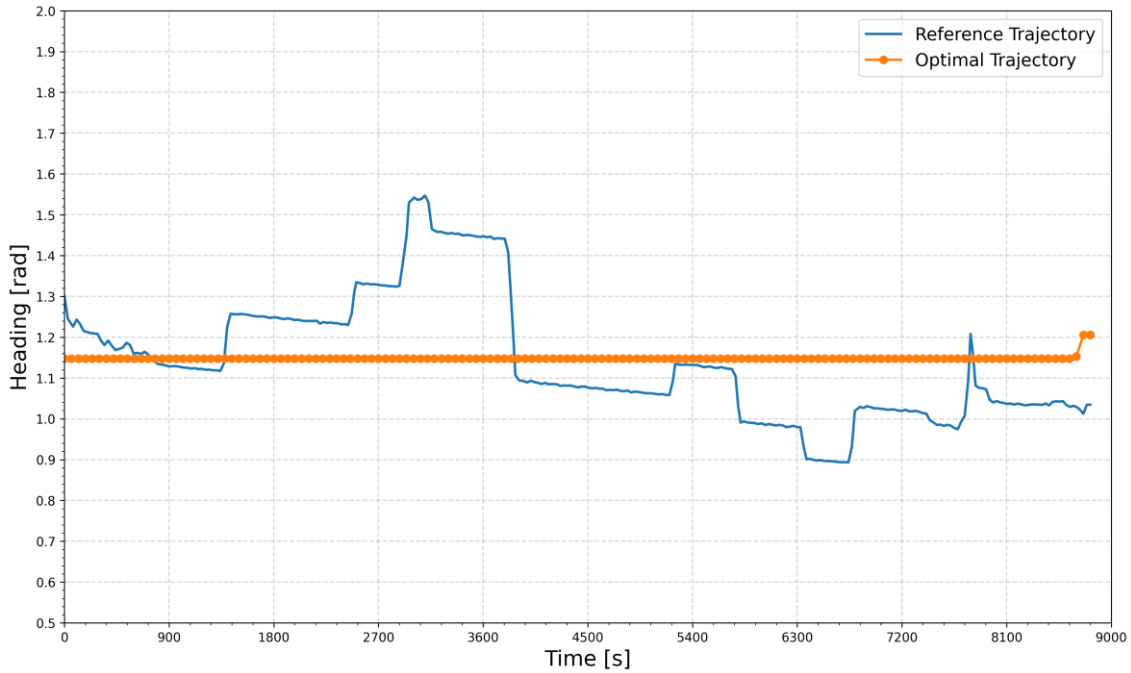


Figure 7.9: Comparison of Heading vs time of reference and green optimal trajectory.

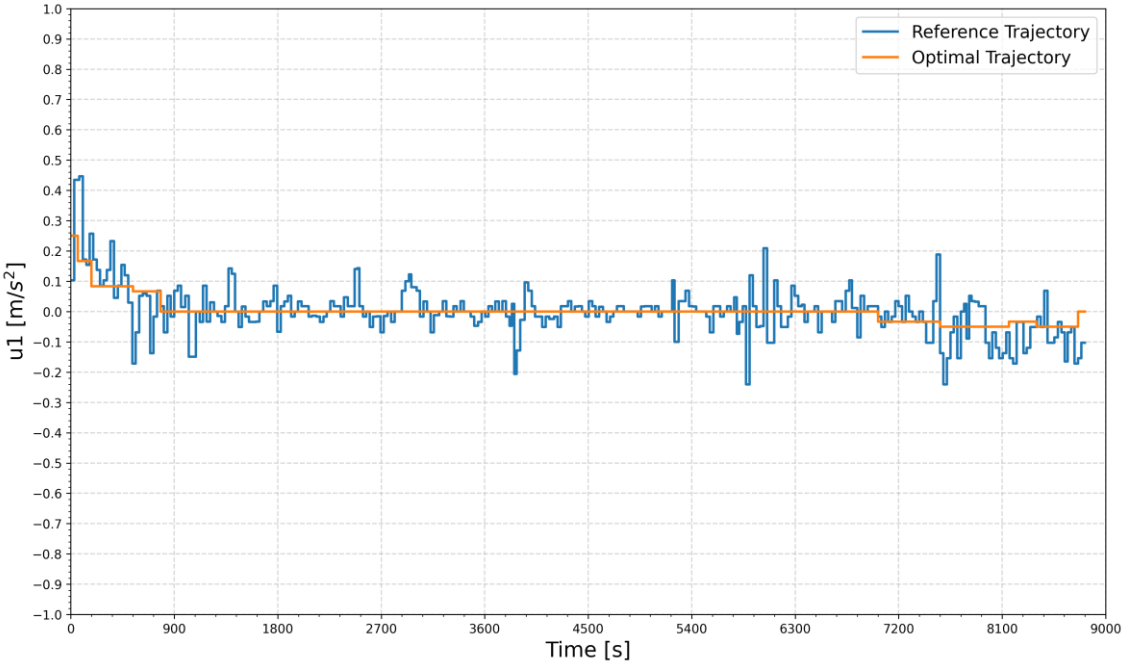


Figure 7.10: Comparison of control u1 vs time of reference and green optimal trajectory.

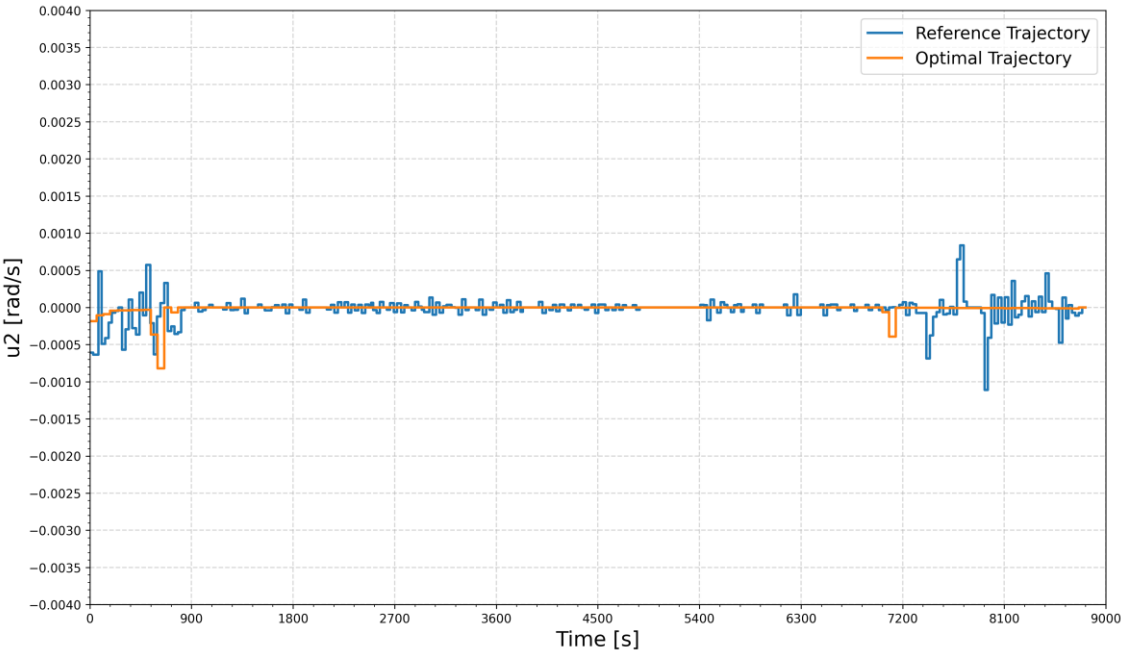


Figure 7.11: Comparison of control u2 vs time of reference and green optimal trajectory.

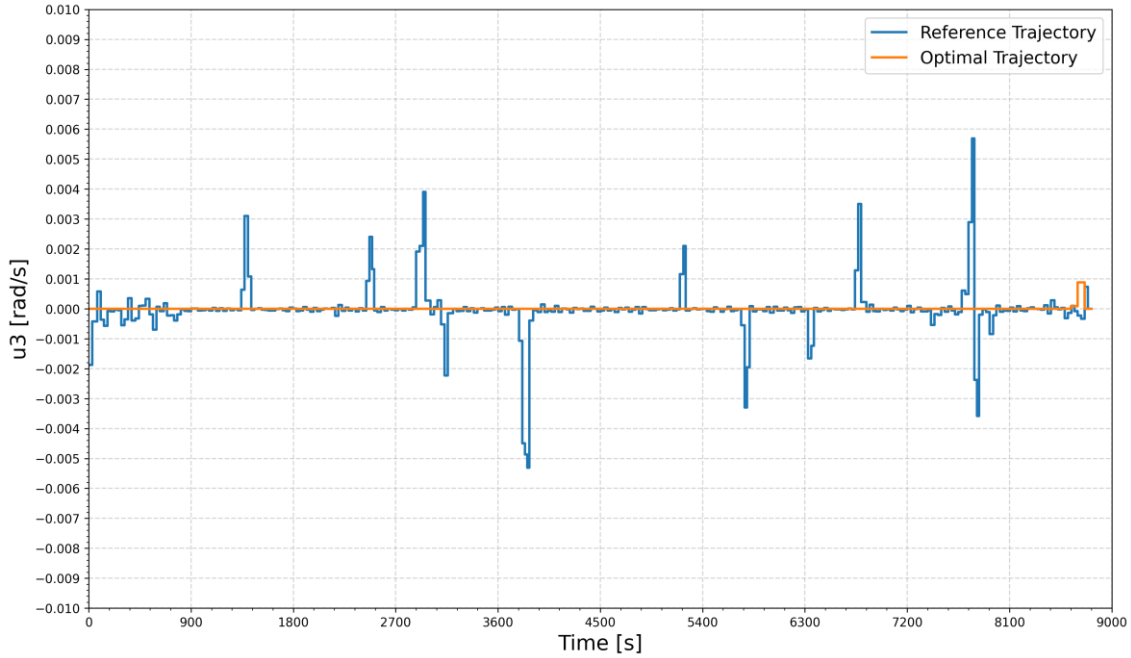


Figure 7.12: Comparison of control u3 vs time of reference and green optimal trajectory.

Figs. (7.10 – 7.12) represent respectively the time history of controls acceleration u1 in $\left[\mathrm{m/s^2}\right]$, flight path angle rate u2 in $\left[\mathrm{rad/s}\right]$, and heading angle rate u3 in $\left[\mathrm{rad/s}\right]$ of the reference and green optimal trajectories.

Table 7.2: Green cost of reference and green optimal trajectory.

	Reference Trajectory [€]	**Optimal Trajectory [€]**	**cost reduction [€]**
Green Cost	**4681.233**	**4196.054**	**485.179**

The aircraft green cost of the reference and optimal trajectory is presented in table (7.2). Based on the results from the table, the aircraft emissions are 485.179 [€] less in optimal trajectory, which is approximately a 10.36% reduction of aircraft green cost by flying the optimal trajectory.

7.5 Summary

This chapter employs the Radial Basis Function (RBF) neural network to predict the fuel flow rate in the climb, cruise, and descent phases of flight. A commercial twinjet subsonic aircraft was considered to design the fuel flow rate model, the true airspeed and flight altitude were taken as the input parameters and the fuel flow rate as the output parameter. The RBF neural network produced a highly accurate fuel flow rate model with the coefficients of determination extremely close to 1 for all three phases of flight. Considering these high coefficients of determination, together with the low relative approximation errors, it can be concluded that the RBF neural network is extremely accurate to predict the fuel flow rate.

This chapter also solves a 4D trajectory optimization problem by optimizing aircraft green cost by applying Modified Dynamic Programming (MDP). The fuel flow rate and the emissions gases of the problem were determined using the fuel flow rate model that was predicted by the RBF neural network. The optimized green trajectory was then compared with the corresponding reference airliner trajectories to quantify the benefit of green cost reduction. The results suggest that the optimal trajectory reduces the green cost by 10.36% than the corresponding reference trajectory.

When actual flight data or manual data from an aircraft are accessible, the RBF neural network could be used to accurately model the fuel flow rate predictions in actual applications.

Chapter 8

8. Conclusions, Contributions, and Future Works

8.1 Conclusions

This thesis focused on generating commercial aircraft optimal trajectories that improve flight efficiency and reduce the environmental impact caused by the airliners. In order to accomplish this goal, Trajectory Optimization Problems (TOPs) need to be formulated and solved by efficient numerical methods. Firstly, a brief description of the three key steps of formulation of the aircraft trajectory optimization problems (e.g., mathematical model, Performance Index, and boundary and path constraints) are presented. Then a brief review of the three basic classes (e.g., indirect methods, direct methods, and dynamic programming) of numerical methods to solve the trajectory optimization problems and their characteristics, advantages, and disadvantages are also discussed.

After the formulation of the trajectory optimization problem and numerical methods to solve them are described. A total of four different problems are considered and solved to generate optimal aircraft trajectories. The first problem considers generating optimal trajectories from predefined 4D waypoint networks. A single source shortest path algorithm was applied to generate the optimal aircraft trajectories that minimize fuel burn and aircraft total trip time between the initial and final waypoint in the networks. The second problem considers generating minimum length optimal trajectory along with a set of predefined 4D waypoints. The cubic spline parameterization was applied to transform the original problem into Nonlinear Programming (NLP) problem, which is then solved numerically using a well-established NLP solver.

The third problem considers generating optimal trajectories between the initial and final 4D waypoints. Dynamic Programming (DP) a well-established numerical method was considered to solve this problem. The DP was chosen because of its several great features to solve aircraft TOP as follow:

- Easy to handle discrete-time dynamic systems with nonlinear characteristics.
- Easy to handle equality and inequality constraints of the system.
- The 4D waypoints representation of the flight trajectory is similar to the discretization of the states grid system of DP.
- The stage-wise optimization procedure of DP is ideally suited for computers.

However, the computational space and time complexity demanded by the traditional DP coupled with an interpolation problem prevent the use of the DP in many practical real-time implementations, especially with the high dimension problems. To overcome the shortcomings of traditional DP a Modified Dynamic Programming (MDP) approach has been proposed. Afterward, the proposed MDP was successfully applied to generate an optimal trajectory between two waypoints that reduces the aircraft fuel consumption and emissions.

The fourth problem considers predicting the aircraft fuel flow rate when actual flight data are available. A Radial Basis Function (RBF) neural network was applied to predict the fuel flow rate in the climb, cruise, and descent phases of flight. Afterward, the predicted fuel flow rate model was used to optimize aircraft green cost.

According to the results of these four problems, the findings are presented below.

Optimal Trajectory Generation from Predefined 4D Waypoint Networks.

- A trajectory optimization tool was developed by using Dijkstra's shortest path algorithm, to find optimal trajectories from a predefined 4D waypoint network.
- The analysis results show promising potential for reduction of fuel consumption and travel time. The results suggest that by flying fuel and time-optimal trajectory for a short-haul flight, it is possible to save 2.4-4.1% of fuel burn and 2.4 - 3.2 % of total travel time. In medium-haul flight by flying the fuel optimal trajectory can potentially save 2.1-2.3% fuel burn and 1.9 – 2.6% of total travel time. For long-haul flights, it is possible to save 2.8-2.9% on fuel burn, and 2.7 -2.9% of total travel time was saved by flying the time-optimal trajectory.
- The developed trajectory optimization tool guarantees global optimal within the considered 4D waypoint networks.

Spline Parameterization Based Nonlinear Trajectory Optimization Along 4D Waypoints.

- In this chapter, a cubic spline approximation method is presented to generate minimum length optimal trajectory along predefined 4D waypoints.
- The Trajectory Optimization Problem (TOP) was solved by a direct optimal control approach based on cubic spline parameterization of the state, its derivatives, and control vector, which transformed the original infinite-dimensional TOP to finite-dimensional Nonlinear Programming (NLP) problem, then the resulting NLP was solved numerically by well-established NLP solver.
- Numerical examples were presented to validate the method. A take-off phase and cruise phase of flight have been considered to generate a minimum length trajectory. The results

suggest that the method was able to reduce the total length by 0.55% for the take-off phase and 0.95% for the cruise phase than the corresponding original nonoptimal trajectories.

- The developed method generated a smooth 4D optimal trajectory with accuracy, which not only ensures the continuity for the trajectory but also for its derivatives.
- The proposed method has the advantage of requiring less computation space and time, which means it can be used successfully to solve an online 4D trajectory optimization problem with high precision.

4D Flight Trajectory Optimization by Modified Dynamic Programming.

- This chapter proposes a modified approach of traditional Dynamic Programming (DP) to solve flight Trajectory Optimization Problems (TOP), in order to cope with the two serious drawbacks of traditional DP.
- The Modified Dynamic Programming (MDP) reduces the first drawback of the curse of dimensionality by limiting the search space at each stage and considering only the grid points of that reduced search space.
- The second drawback of the menace of the expending grid is also solved by the MDP approach by generating the control values inside the allowable range. This generation of the control values guarantees reachable grid points for the states at any stage.
- The proposed MDP approach has been validated by a couple of case studies that generate optimal trajectories that reduce aircraft fuel consumption and emissions. The numerical examples consist of the climb phase, cruise phase, descent phase, and global trajectory. In all examples, the proposed MDP approach was able to successfully generate optimal trajectories that reduce aircraft fuel consumption or emissions.
- The MDP approach significantly reduces the computation efforts of Traditional DP to solve the TOP, which implies it can also be applied to an online trajectory generation problem.

Fuel Flow Rate Prediction Using Radial Basis Function Neural Network.

- This chapter employs the Radial Basis Function (RBF) neural network to predict the fuel flow rate in the climb, cruise, and descent phases of flight. the true airspeed and flight altitude were taken as the input parameters and the fuel flow rate as the output parameter.
- The RBF neural network was able to produce a highly accurate fuel flow rate model with the high value of coefficients of determination together with the low relative approximation errors for all three phases of flight.
- This chapter also solves a 4D trajectory optimization problem by optimizing aircraft green cost by using the fuel flow rate model that was predicted by the RBF neural network. The

results suggest that the optimal trajectory reduces the green cost by 10.36% than the corresponding reference trajectory.

8.2 Contributions

The main contributions of this thesis are:

- The application of a single source shortest path algorithm to generate optimal trajectories from predefined 4D waypoint networks. The optimal aircraft trajectories were generated to minimize the fuel burn and aircraft total trip time between the initial and final waypoint in the networks.
- The application of Cubic Spline Interpolation (CSI) to generate minimum length trajectory along with predefined 4D waypoints. The original Trajectory Optimization Problem (TOP) is transformed into a Nonlinear Programming (NLP) problem by the CSI, then an NLP solver is used to solve the resulting NLP and to determine the minimum length optimal trajectory.
- The proposal of a modified approach of the Dynamic Programming (DP) method to solve the aircraft TOP. The proposed Modified Dynamic Programming (MDP) approach not only reduces the computational burden of the traditional DP by reducing the number of grid points that need to consider in each stage but also resolves the limitation of the menace of the expanding grid problem while retaining the appealing features of DP. This allows the MDP to be applied in high dimension problems such as 4D flight TOPs. The MDP approach is successfully validated by generating optimal trajectories that reduce fuel consumption and aircraft emissions.
- The application of Radial Basis Function (RBF) neural network to predict the fuel flow rate of a commercial aircraft for the climb, cruise, and descent phases of flight. The predicted fuel flow rate model was used to optimize aircraft green cost.

8.3 Future Works

Several future works of this thesis can be identified. First, the results presented in the thesis consider no wind condition. Thus, by considering the dynamic weather information this work can be extended. Secondly, the air traffic regulation constraints can be imposed to extend this work. Moreover, other constraints can be considered, such as avoidance of obstacles or restricted areas to generate more realistic commercial trajectories. Thirdly, the work can be extended by considering other performance indexes to reduce the environmental impact caused by the airlines, such as noise emissions.

Bibliography

[1] L. Jensen, R. Hansman, J. Venuti, and T. Reynolds, "Commercial airline speed optimization strategies for reduced cruise fuel consumption," *AIAA 2013 Aviation Technology, Integration, and Operations Conference*, Los Angeles, CA, USA, 2013. DOI: 10.2514/6.2013-4289.

[2] J. T. Betts, "Survey of numerical methods for trajectory optimization, " *Journal of Guidance, Control and Dynamics*, vol. 21, no. 2, pp. 193-207, 1998. DOI: 10.2514/2.4231.

[3] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no.1, pp.497-528, 2009.

[4] J. W. Burrows, "Fuel-optimal aircraft trajectories with fixed arrival times," *Journal of Guidance, Control, and Dynamics*, vol. 6, no. 1, pp. 14-19, 1983. DOI: 10.2514/3.19796.

[5] F. Neuman, and E. Kreindler, "Minimum-fuel, three-dimensional flight paths for jet transports, " *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 650-657, 1985. DOI: 10.2514/3.20035.

[6] A. Chakravarty, "Four-dimensional fuel-optimal guidance in the presence of winds," *Journal of Guidance, Control and Dynamics*, vol. 8, no.1, pp. 16-22, 1985. DOI: 10.2514/3.19929.

[7] R.L. Schultz, "Three-dimensional trajectory optimization for aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 936-943, 1990. DOI: 10.2514/3.20564.

[8] R. L. Barron, and C. M. Chick III, "Improved indirect method for air-vehicle trajectory optimization," *Journal of Guidance, Control and Dynamics*, vol. 29, no. 3, pp. 643–652, 2006. DOI: 10.2514/1.16228.

[9] C. Jansch, and M. Paus, "Aircraft trajectory optimization with direct collocation using movable gridpoints," *1990 American Control Conference*, San Diego, CA, USA, 1990, pp. 262-267. DOI: 10.23919/ACC.1990.4790738.

[10] T. Raivio, H. Ehtamo, and R. P. Hämäläinen, "Aircraft trajectory optimization using nonlinear programming," *System Modelling and Optimization*, Springer, pp. 435-441, 1996. DOI: 10.1007/978-0-387-34897-1_52.

[11] K. Ahmed, K. Bousson, and M. Coelho, "Spline parameterization based nonlinear trajectory optimization along 4D waypoints," *Advances in Aircraft and Spacecraft Science*, vol. 6, no. 5, pp. 391-407, 2019. DOI: 10.12989/aas.2019.6.5.391.

[12] K. Bousson, and T. A. Gameiro, "A quintic spline approach to 4D trajectory generation for unmanned aerial vehicles," *International Review of Aerospace Engineering (IREASE)*, vol. 8, no.1, pp. 1-9, 2015. DOI: 10.15866/irease.v8i1.4780.

[13] M. Soler, A. Olivares, and E. Staffetti, "Multiphase optimal control framework for commercial aircraft four-dimensional flight-planning problems," *Journal of Aircraft*, vol. 52, no. 1, pp. 274-286, 2015. DOI: 10.2514/1.C032697.

[14] Y. Tian, X. He, Y. Xu, L. Wan, and B. Ye, "4D Trajectory Optimization of Commercial Flight for Green Civil Aviation," *IEEE Access*, vol. 8, pp. 62815-62829, 2020. DOI: 10.1109/ACCESS.2020.2984488.

[15] M. Houacine, and S. Khardi, "Gauss pseudospectral method for less noise and fuel consumption of aircraft operations," *Journal of Aircraft*, vol. 47, no. 6, pp. 2152-2159, 2010. DOI: 10.2514/1.C031007.

[16] M. Bittner, F. Fisch, and F. Holzapfel, "A multi-model gauss pseudospectral optimization method for aircraft trajectories," *AIAA Atmospheric Flight Mechanics Conference*, Minneapolis, Minnesota, 2012. DOI: 10.2514/6.2012-4728.

[17] K. Bousson, and P. Machado, "4D flight trajectory optimization based on pseudospectral methods," *International Journal of Aerospace and Mechanical Engineering*, vol. 4, no.9, pp. 879 -885, 2010. DOI: 10.5281/zenodo.1076520.

[18] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, USA, 1957.

[19] Y. Miyamoto, N. K. Wickramasinghe, A. Harada, Y. Miyazawa, and K. Funabiki, "Analysis of fuel-efficient airliner flight via dynamic programming trajectory optimization," *Transactions of the Japan Society for Aeronautical and Space Sciences,* vol. 11, pp. 93–98, 2013. DOI: 10.2322/tastj.11.93.

[20] M. C. Waller, J. G. Rigopoulos, D. R. Blackman, and T. F. Berreen, "Considerations in the application of dynamic programming to optimal aircraft trajectory generation," *IEEE Conference on Aerospace and Electronics*, Dayton, USA, May 1990, pp. 574-579. DOI: 10.1109/NAECON.1990.112828.

[21] S. Grabbe, B. Sridhar, and N. Cheng, "Central East Pacific Flight Routing," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, 2006. DOI: 10.2514/6.2006-6773.

[22] P. Hagelauer, and F. A. C. Mora-Camino, "A soft dynamic programming approach for on-line aircraft 4D trajectory optimization," *European Journal of Operational Research*, vol. 107, no. 1, pp. 87–95, 1998. DOI: 10.1016/S0377217(97)00221X.

[23] Y. Miyazawa, N. K. Wickramasinghe, A. Harada, and Y. Miyamoto, "Dynamic programming application to airliner four-dimensional optimal flight trajectory," *Proceedings of the AIAA Guidance, Navigation, and Control (GNC) Conference*, Boston, Massachusetts, U.S.A., August 2013. DOI: 10.2514/6.2013-4969.

[24] A. Harada, H. Matsuda, and Y. Miyazawa, "Dynamic programming trajectory optimization by piecewise linear approximation, " *In Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Kissimmee, FL, USA, January 2015. Doi:10.2514/6.2015-1075.

[25] S. Khardi, "Aircraft flight path optimization. The Hamilton-Jacobi-Bellman considerations," *Applied Mathematical Sciences*, Hikari, vol. 6, no. 25, pp. 1221-1249, 2012.

[26] C. Parzani, and S. Puechmorel, "On a Hamilton-Jacobi-Bellman Approach for Coordinated Optimal Aircraft Trajectories Planning," *Optimal Control Applications and Methods*, John Wiley & Sons, 2017. DOI: 10.1002/oca.2389.

[27] R. Luus, "Optimal control by dynamic programming using systematic reduction in grid size," *International Journal of Control,* vol. 51, no. 5, pp. 995-1013, 1990. DOI: 10.1080/00207179008953533.

[28] K. Bousson, "Single gridpoint dynamic programming for trajectory optimization," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, San Francisco, California, USA, August 2005. DOI: 10.2514/6.2005-5902.

[29] A. Miele, "General solutions of optimum problems in nonstationary flight," NACA-TM-1388, National Advisory Committee for Aeronautics; Washington, DC, USA, 1955.

[30] G. Serafino, "Multi-objective Aircraft Trajectory Optimization for Weather Avoidance and Emissions Reduction," *In International Workshop on Modelling and Simulation for Autonomous Systems*, Springer, Cham, Switzerland, 2015. DOI: 10.1007/978-3-319-22383-4_18.

[31] H. G. Visser, and R. A. A. Wijnen, "Optimization of noise abatement departure trajectories," *Journal of Aircraft,* vol. 38, no. 4, pp. 620–627, 2001. DOI: 10.2514/2.2838.

[32] N. T. Ho, and J. P. Clarke, "Methodology for optimizing parameters of noise-abatement approach procedures," *Journal of Aircraft,* vol. 44, no. 4, pp. 1168-1176, 2007. DOI: 10.2514/1.22292.

[33] X. Prats, V. Puig, and J. Quevedo, "Equitable aircraft noise-abatement departure procedures," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 1, pp. 192–203, 2011. DOI: 10.2514/1.49530.

[34] D. McConnachie, P. Bonnefoy, and A. Belle, "Investigating benefits from continuous climb operating concepts in the national airspace system," *Eleventh USA/Europe Air Traffic Management Research and Development Seminar (ATM2015)*, Lisbon, Portugal, June 2015.

[35] J. Rosenow, S. Förster, and H. Fricke, "Continuous climb operations with minimum fuel burn," *6th SESAR Innovation days*, Netherlands, November 2016.

[36] M. V. Díaz, V. F. G. Comendador, J. G. Carretero, and R. M. A. Valdés, "Environmental benefits in terms of fuel efficiency and noise when introducing continuous climb operations as part of terminal airspace operation," *International Journal of Sustainable Transportation,* 2019. DOI: 10.1080/15568318.2019.1651924.

[37] J. L. Speyer, "On the fuel optimality of cruise," *Journal of Aircraft*, vol. 10, no. 11, pp. 763–765, 1973. DOI: 10.2514/3.60304.

[38] R. L. Schultz, "Fuel optimality of cruise," *Journal of Aircraft*, vol. 11, no. 9, pp. 586-587, 1974. DOI: 10.2514/3.60391.

[39] M. L. Shepard, and K. D. Bilimoria, "Optimization of aircraft cruise performance," *AIAA 16th Atmospheric Flight Mechanics Conference*, Boston, USA, August 1989, pp. 338-347. DOI: 10.2514/6.1989-3386.

[40] S. Liden, "Optimum cruise profiles in the presence of winds," *Proceedings IEEE/AIAA 11th Digital Avionics Systems Conference*, Seattle, USA, October 1992, pp. 254-261. DOI: 10.1109/DASC.1992.282147.

[41] D. M. Pargett, and M. D. Ardema, "Flight path optimization at constant altitude," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1197-1201, 2007. DOI: 10.2514/1.28954.

[42] A. Franco, D. Rivas, and A. Valenzuela, "Minimum-fuel cruise at constant altitude with fixed arrival time," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 280–285, 2010. DOI: 10.2514/1.46465.

[43] A. Franco, and D. Rivas, "Minimum-cost cruise at constant altitude of commercial aircraft including wind effects," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1253–1260, 2011. DOI: 10.2514/1.53255.

[44] L. Delgado, and X. Prats, "Fuel consumption assessment for speed variation concepts during the cruise phase," Conference on Air Traffic Management Economics, Belgrade, Serbia, September 2009.

[45] M. Pawlak, A. Majka, M. Kuzniar, and J. pawluczy, "Model of emission of exhaust compounds of jet aircraft in cruise phase enabling trajectory optimization," *Transport*, vol. 35, no. 1, pp. 87-97, 2020. DOI: 10.3846/transport.2020.12243.

[46] Y. Tian, L. Wan, B. Ye, and D. Xing, "Cruise Flight Performance Optimization for Minimizing Green Direct Operating Cost," *Sustainability*, vol. 11, 3899, 2019. DOI: 10.3390/su11143899.

[47] L. E. Fogarty, and R. M. Howe, "Trajectory optimization by a direct descent process," *SIMULATION*, vol. 11, no. 3, pp. 145-155, 1968. DOI: 10.1177/003754976801100308.

[48] S. G. Park, and J. P. Clarke, "Vertical trajectory optimization to minimize environmental impact in the presence of wind," *Journal of Aircraft*, vol. 53, no. 3, pp. 725-737, 2016. DOI: 10.2514/1.C032974.

[49] F. J. M. Wubben, and J. J. Busink, "Environmental benefits of continuous descent approaches at Schiphol airport compared with conventional approach procedures," NLR-TP-2000-275, National Aerospace Laboratory NLR, Netherlands, 2000.

[50] S. G. Park, and J. P. Clarke, "Optimal control based vertical trajectory determination for continuous descent arrival procedures," *Journal of Aircraft*, vol. 52, no. 5, pp. 1469-1480, 2015. DOI: 10.2514/1.C032967.

[51] Y. Cao, T. Kotegawa, and J. Post, "Evaluation of continuous descent approach as a standard terminal airspace operation," *9th USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)*, Berlin, Germany, June 2011.

[52] Y. Tian, L. Wan, K. Han, and B. Ye, "Optimization of terminal airspace operation with environmental considerations," *Transportation Research Part D Transport and Environment,* vol. 63, pp. 872–889, 2018. DOI: 10.1016/j.trd.2018.06.018.

[53] J. F. Barman, and H. Erzberger, "Fixed-range optimum trajectories for short-haul aircraft," *Journal of Aircraft*, vol. 13, no. 10, pp. 748-754, 1976. DOI: 10.2514/3.58706.

[54] H. Erzberger, and H. Lee, "Constrained optimum trajectories with specified range," *Journal of Guidance, Control and Dynamics*, vol. 3, no. 1, pp. 78-85, 1980. DOI: 10.2514/3.55950.

[55] D. Rivas, A. Valenzuela, and J. L. de Augusto, "Computation of global trajectories of commercial transport aircraft," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 227, no. 1, pp. 142-158, 2012. DOI: 10.1177/0954410011427107.

[56] R. Patrón, Y. Berrou, and R. Botez, "Climb, cruise and descent 3D trajectory optimization algorithm for a flight management system," *AIAA/3AF Aircraft Noise and Emissions Reduction Symposium*, Atlanta, GA, June 2014. DOI: 10.2514/6.2014-3018.

[57] C. Celis, V. Sethi, D. Zammit-Mangion, R. Singh, and P. Pilidis, "Theoretical optimal trajectories for reducing the environmental impact of commercial aircraft operations," *Journal of Aerospace Technology and Management*, vol. 6, no. 1, pp. 29–42, 2014. DOI: 10.5028/jatm.v6i1.288.

[58] K. Ahmed, and K. Bousson, "Generating time optimal trajectory from predefined 4D waypoint networks," *International Review of Aerospace Engineering (IREASE)*, vol. 10, no. 4, pp. 207-214, 2017. DOI: 10.15866/irease.v10i4.12570.

[59] K. Ahmed, M. Coelho, and K. Bousson, "Optimal fuel saving in 4D waypoint networks," *International Congress on Engineering — Engineering for Evolution,* KnE Engineering, pp. 500–513, 2020. DOI 10.18502/keg.v5i6.7065.

[60] A. Miele, *Flight Mechanics: Theory of Flight Paths*, Dover Publication, New York, USA, 2016.

[61] N. X. Vinh, *Flight Mechanics of High-Performance Aircraft*, Cambridge University Press, Cambridge, England, 1993.

[62] R. Slattery, and Y. Zhao, "Trajectory synthesis for air traffic automation, "*Journal of Guidance, Control, and Dynamics*, vol. 20, no. 2, pp. 232–238, 1997. DOI: 10.2514/2.4056.

[63] D. G. Hull, *Fundamentals of Airplane Flight Mechanics*, Springer, Berlin, Germany, 2007.

[64] EUROCONTROL, "User manual for the Base of Aircraft Data (BADA) Revision 3.11," Eurocontrol Experimental Centre (EEC) Technical/Scientific Report No. 13/04/16-01, Brétigny-sur-Orge, France, 2013.

[65] D. J. Sutkus, S. L. Baughcum, and D. P. DuBois, "Scheduled Civil Aircraft Emission Inventories for 1999: Database Development and Analysis," NASA/CR-2001-21121, NASA, 2001.

[66] O. J. Hadaller, and A. M. Momenthy, "The characteristics of future fuels," Boeing publication D6 -54940, 1989.

[67] S. L. Baughcuma, T. G. Tritz, S. C. Henderson, and D. C. Pickett, "Scheduled civil aircraft emission inventories for 1992: Database development and analysis," Project Report NASA CR 4700, NASA, 1996.

[68] O. Bolza, "Über den anormal fall beim lagrangeschen una mayerschen problem mit gemischten bedingungen und variablen endpunkten," *Mathematische Annalen*, vol. 74, pp. 430-446, 1913.

[69] G. A. Bliss, "The problem of Bolza in the calculus of variations," *Annals of Mathematics*, vol. 33, no. 2, pp. 261-274, 1932. DOI: 10.2307/1968328.

[70] B. Roberson, "Fuel Conservation Strategies: Cost Index Explained," Boeing, pp. 26-28, 2007.

[71] A. E. Bryson, and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation and, Control*, Taylor & Francis, New York, USA, 1975.

[72] D. E. Kirk, *Optimal Control Theory: An Introduction*, Dover Publication, Inc., New York, USA, 2004.

[73] D. S. Naidu, *Optimal Control Systems*, CRC Press, Florida, USA, 2003.

[74] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*, John Wiley & Sons, New York, USA, 1962.

[75] M. A. Athans, and P.L. Falb, *Optimal Control: An Introduction to The Theory and Its Applications*, Dover Publications, Mineola, New York, USA, 2006.

[76] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Second Ed., SIAM Press, Philadelphia, USA, 2010.

[77] O. V. Stryk, and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, no. 1, pp. 357-373, 1992. DOI: 10.1007/BF02071065.

[78] D. G. Hull, "Conversion of optimal control problems into parameter optimization problems," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 57-60, 1997. DOI: 10.2514/2.4033.

[79] M. S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Wiley-Interscience, New Jersey, USA, 2006.

[80] D. E. Kirk, "An introduction to dynamic programming," *IEEE Transactions on Education*, vol. 4, no. 10, pp. 212-219, 1967. DOI: 10.1109/TE.1967.4320291.

[81] J. Stoer, and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, USA, 2002.

[82] H. B. Keller, *Numerical Methods for Two-Point Boundary Value Problems*, Blaisdell Publishing Company, Waltham, Massachusetts, USA, 1968.

[83] E. D. Dickmanns, and K.H. Well, "Approximate solution of optimal control problems using third-order Hermite polynomial functions," *Proceedings of the 6th Technical Conference on Optimization Techniques*, Vol. IFIP-TC7. Springer-verlag, New York, 1975.

[84] E. D. Dickmanns, "Efficient convergence and mesh refinement strategies for solving general ordinary two-point boundary value problems by collocated hermite approximation," *2nd IFAC Workshop on Optimisation*, Oberpfaffenhofen, Germany, 1980.

[85] D. Bertsekas, *Nonlinear Programming*, Athena Scientific Publishers, Belmont, Massachusetts, USA, 2004.

[86] J. Z. Ben-Asher, *Optimal Control Theory with Aerospace Applications*, American Institute of Aeronautics and Astronautics, USA, 2010.

[87] P. E. Gill, W. Murray, M. A. Saunders, and M.H. Wright, "User's guide for NPSOL (Version 4.0): A FORTRAN Package for Nonlinear Programming," Department of Operations Research, Stanford University, USA, 1986.

[88] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Review*, vol. 47, no.1, pp. 99–131, 2005. DOI: 10.1137/S0036144504446096.

[89] R. H. Byrd, J. Nocedal, and R. A. Waltz, "KNITRO: An integrated package for nonlinear optimization," *Large Scale Nonlinear Optimization*, Springer Verlag, vol. 83, pp. 35–59, 2006. DOI: 10.1007/0-387-30065-1_4.

[90] A. Wächter, and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006. DOI: 10.1007/s10107-004-0559-y.

[91] R. A. Bartlett, and L. T. Biegler, "rSQP++ : An object-oriented framework for successive quadratic programming," *(eds) Large-Scale PDE-Constrained Optimization*, Springer, Berlin, vol. 30, pp. 316–330, 2003. DOI: 10.1007/978-3-642-55508-4_19.

[92] B. A. Conway, "A Survey of methods available for the numerical optimization of continuous dynamic systems," *Journal of Optimization Theory and Applications*, vol. 152, pp. 271–306, 2012. DOI: 10.1007/s10957-011-9918-z.

[93] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Longman Publishing Co., USA, 1989.

[94] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific publishing co, New York, USA, 1999.

[95] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. DOI: 10.1126/science.220.4598.671.

[96] J. A. Kennedy, and R. C. Eberhardt, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks,* Perth, WA, Australia, vol. 4, pp. 1942–1948, 1995. DOI: 10.1109/ICNN.1995.488968.

[97] J. A. Kennedy, *Swarm Intelligence*, (eds) Handbook of Nature-Inspired and Innovative Computing. Springer, Boston, MA, 2006.

[98] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," *Proceedings of the European Conference on Artificial Life, ECAL'91*, Elsevier Publishing, Paris, France, pp. 134-142, 1991.

[99] D. Kraft, "On converting optimal control problems into nonlinear programming problems," *Computational Mathematical Programming*, vol. 15, pp. 261–280, 1985. DOI: 10.1007/978-3-642-82450-0_9.

[100] C. A. Kluever, "Optimal low-thrust interplanetary trajectories by direct method techniques," *The Journal of the Astronautical Science*, Vol. 41, no. 3, pp. 247-262, 1997.

[101] H. G. Bock, and K. J. Plitt, "A multiple shooting algorithm for direct solution optimal control problems," *Proceedings of the 9th IFAC World Congress*, Pergamon Press, Budapest, pp. 243-247, 1984.

[102] T. Tsang, D. Himmelblau, and T. Edgar, "Optimal control via collocation and nonlinear programming," *International Journal of Control*, vol. 21, no. 5, pp. 763–768, 1975. DOI: 10.1080/00207177508922030.

[103] C. R. Hargraves, and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987. DOI: 10.2514/3.20223.

[104] O. V. Stryk, "Numerical solution of optimal control problems by direct collocation," *Optimal Control - ISNM International Series of Numerical Mathematics*, vol. 111, pp. 129-143, 1993. DOI: 10.1007/978-3-0348-7539-4_10.

[105] P. Williams, "Hermite-Legendre-Gauss-Lobatto direct transcription methods in trajectory optimization," *Advances in the Astronautical Sciences*, vol. 120, no. 1, pp. 465–484, 2005. DOI: 10.2514/1.42731.

[106] G. Elnagar, M. A. Kazemi, and M. Razzaghi, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1793–1796, 1995. DOI: 10.1109/9.467672.

[107] D. A. Benson, G. T. Huntington, T. P. Thorvalden, and A. V. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance, Control and Dynamics*, vol. 29, no. 6, pp. 1435–1440, 2006. DOI: 10.2514/1.20478.

[108] G. T. Huntington, D. Benson, A. V. Rao, "A comparison of accuracy and computational efficiency of three pseudospectral methods," *Guidance, Navigation, and Control Conference and Exhibit*, South Carolina, USA, 2007. DOI: 10.2514/6.2007-6405.

[109] F. Fahroo, and I.M. Ross, "Direct trajectory optimization by a Chebyshev pseudospectral method," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 160-166, 2002. DOI: 10.2514/2.4862.

[110] K. Bousson, "Chebyshev pseudospectral trajectory optimization of differential inclusion models," *SAE World Aviation Congress*, Montreal, Canada, paper no. 2003-01-3044, 2003. DOI: 10.4271/2003-01-3044.

[111] K. Bousson, and P. Machado, "4D trajectory generation and tracking for waypoint-based aerial navigation," *WSEAS Transactions on Systems and Control*, vol. 8, no. 3, pp. 105–119, 2013.

[112] X. Guo, and M. Zhu, "Direct trajectory optimization based on a mapped Chebyshev pseudospectral method," *Chinese Journal of Aeronautics*, vol. 26, no. 2, pp. 401-412, 2013. DOI: 10.1016/j.cja.2013.02.018.

[113] F. Fahroo, and I. M. Ross, "Costate estimation by a Legendre pseudospectral," *Journal of Guidance, Control and Dynamics*, vol. 24, no. 2, pp. 270-277, 2001. DOI: 10.2514/2.4709.

[114] J. Zhao, R. Zhou, and X. Jin, "Gauss pseudospectral method applied to multi-objective spacecraft trajectory optimization," *Journal of Computational and Theoretical Nanoscience*, vol. 11, no. 10, pp. 2242–2246, 2014. DOI: 10.1166/jctn.2014.3685.

[115] L. Ma, Z. Shao, W. Chen, X. Lv, and Z. Song, "Three-dimensional trajectory optimization for lunar ascent using Gauss pseudospectral method," *Guidance, Navigation, and Control Conference*, San Diego, California, USA, 2016. DOI: 10.2514/6.2016-1372.

[116] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010. DOI: 10.1016/j.automatica.2010.06.048.

[117] D. Garg, W. W. Hager, and A. V. Rao, "Pseudospectral methods for solving infinite-horizon optimal control problems," *Automatica*, vol. 47, no. 4, pp. 829–837, 2011. DOI: 10.1016/j.automatica.2011.01.085.

[118] H. Seywald, "Trajectory optimization based on differential inclusion," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 3, pp. 480–487, 1994. DOI: 10.2514/3.21224.

[119] R. R. Kumar, and H. Seywald, "Should controls be eliminated while solving optimal control problems via direct methods?," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 2, pp. 418–423, 1996. DOI: 10.2514/3.21634.

[120] D. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific Publishers, Vol. 1, 3rd ed. Belmont, Massachusetts, USA, 2005.

[121] L. Lapidus, and R. Luus, *Optimal Control of Engineering Processes*, Waltham, Blaisdell Publishing Co, 1967.

[122] R. Luus, "Dynamic Programming: Optimal Control Applications," *Encyclopedia of Optimization*, Springer, Boston, MA, 2008. DOI: 10.1007/978-0-387-74759-0_151.

[123] R. Luus, *Iterative Dynamic Programming,* Chapman and Hall / CRC, London, UK, 2000.

[124] R. Luus, "Application of dynamic programming to high-dimensional non-linear optimal control problems," *International Journal of Control,* vol. 52, no. 1, pp. 239-250, 1990. DOI: 10.1080/00207179008953533.

[125] R. Luus, and S. G. smith, "Application of dynamic programming to high-dimensional systems described by difference equations," *Chemical Engineering Technology,* vol. 14, no. 2, pp. 122-126, 1991. DOI: 10.1002/ceat.270140210.

[126] A. A. Seemkooei, "Comparison of different algorithm to transform geocentric to geodetic coordinates," *Survey Review*, vol. 36, no. 286, pp. 627-633, 2002. DOI: 10.1179/003962602791482966.

[127] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959. DOI: 10.1007/BF01386390.

[128] E.V. Denardo, *Dynamic Programming: Models and Applications*, Dover Publications, Mineola, New York, USA, 2003.

[129] M. Sniedovich, "Dijkstra's algorithm revisited: the dynamic programming connexion," *Journal of Control and cybernetics*, vol. 35, no. 3, pp. 599-620, 2006.

[130] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, *Algorithms*, McGraw-Hill, New York, USA, 2006.

[131] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, London, England, 2009.

[132] C. Hart, "Graph Theory Topics in Computer Networking," University of Houston-Downtown, Department Computer and Mathematical Sciences, Senior Project, pp. 13-20, 2013.

[133] R. L. Burden, and J. D. Faires, *Numerical Analysis* (Ninth Ed.), Cengage Learning, Boston, USA, 2011.

[134] Z. Wang, and J. Ouyang, "Curve length estimation based on cubic spline interpolation in gray-scale images," *Journal of Zhejiang University-SCIENCE C (Computers & Electronics)*, vol. 14, no. 10, pp. 777-784, 2013. DOI: 10.1631/jzus.C1300056.

[135] M. Marsden, "Cubic spline interpolation of continuous functions," *Journal of Approximation Theory*, vol. 10, no. 2, pp. 103-111, 1974. DOI: 10.1016/0021-9045(74)90109-9.

[136] K. Ahmed, K. Bousson, and M. Coelho, "4D flight trajectory optimization by modified dynamic programming," *Aerospace Europe Conference 2020 (AEC2020),* Bordeaux, France, February 2020.

[137] J. H. Seinfeld, and L. Lapidus, "Aspects of forward dynamic programming algorithm," *Industrial & Engineering Chemistry Process Design and Development*, vol. 7, no. 3, pp. 475-478, 1968. DOI: 10.1021/i260027a027.

[138] S. Verdu, and H. V. Poor, "Backward, forward and backward-forward dynamic programming models under commutativity conditions," *In Proceedings of 23rd IEEE conference decision control*, Las Vegas, NV, USA, pp. 1081-1086, December 1984.

[139] A. A. Sitinjak, E. Pasaribu, J. E. Simarmata, T. Putra, and H. Mawengkang, "The analysis of forward and backward dynamic programming for multistage graph," *IOP Conference Series: Materials Science and Engineering*, vol. 300, no. 012010, IOP Publishing, 2018. DOI: 10.1088/1757-899X/300/1/012010.

[140] EUROCONTROL, "Aircraft performance summary tables for the Base of Aircraft Data (BADA) Revision 3.4," Eurocontrol Experimental Centre (EEC) Note No. 06/02, Brétigny-sur-Orge, France, 2002.

[141] K. Ahmed, K. Bousson, and M. Coelho, "A modified dynamic programming approach for 4D minimum fuel and emissions trajectory optimization," Aerospace, vol. 8, no. 5: 135, 2021. DOI: 10.3390/aerospace8050135.

[142] D. Graupe, *Principles of artificial neural networks*, 2nd Edition Vol. 6. World Scientific, 2007.

[143] K. Mehrotra, C. K. Mohan, and S. Ranka, *Elements of artificial neural networks*, MIT Press, 1997.

[144] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural network design*. PWS Publishing Co, 1996.

[145] D. S. Broomhead, and D. Lowe, "Multivariable function interpolation and adaptive networks," *Complex Systems*, vol. 2, no.3, pp. 321–355, 1988.

[146] T. Poggio, and F. Girosi, "A theory of networks for approximation and learning," A.I. Memo No. 1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1989.

[147] J. Moody, and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989. DOI: 10.1162/neco.1989.1.2.281.

[148] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris, and D. M. Hummels. "On the training of radial basis function classifiers," *Neural networks*, vol. 5, no. 4, pp. 595-603, 1992. DOI: 10.1016/S0893-6080(05)80038-3.

[149] T. L. Fine, *Feedforward neural network methodology*, Springer Science & Business Media, 2006.

Annex A

A. Aircraft Performance Operational Data

A.1 Aircraft A1

Aircraft Type

Symbols	Value	Description
n_{eng}	2	Number of engines
Engine type	CFM56-7	Jet
Wake Category	M	Medium

Aircraft Mass

Symbols	Value	Units	Description
m_{ref}	60000	[Kg]	Reference mass
m_{min}	38280	[Kg]	Minimum mass
m_{max}	70800	[Kg]	Maximum mass
m_{pyld}	16920	[Kg]	Maximum payload mass

Flight Envelope

Symbols	Value	Units	Description
V_{MO}	340	[knots]	Maximum operating speed - (CAS)
M_{MO}	0.82	N/A	Maximum operating Mach number
h_{MO}	41000	[ft]	Maximum operating altitude
h_{max}	37700	[ft]	Maximum altitude at MTOW and ISA
G_{t}	-131	[ft/K]	Temperature gradient on maximum altitude

Aerodynamics

Symbols	Value	Units	Description
S	124.65	$[m^2]$	Reference wing surface area
C_{Lbo}	1.29	N/A	Buffet onset lift coefficient
k	0.505	N/A	Buffeting gradient
$V_{stall,\,CR}$	143	[knots]	Stall Speed (cruise) - (CAS)
$C_{D0,\,CR}$	0.0235	N/A	Parasitic drag coefficient (cruise)
$C_{D2,\,CR}$	0.0445	N/A	Induced drag coefficient (cruise)
$V_{stall,\,IC}$	115	[knots]	Stall Speed (initial climb) - (CAS)
$C_{D0,\,IC}$	0.0270	N/A	Parasitic drag coefficient (initial climb)
$C_{D2,\,IC}$	0.0441	N/A	Induced drag coefficient (initial climb)
$V_{stall,\,TO}$	109	[knots]	Stall Speed (take-off) - (CAS)
$C_{D0,\,TO}$	0.0333	N/A	Parasitic drag coefficient (Take-off)
$C_{D2,\,TO}$	0.0428	N/A	Induced drag coefficient (Take-off)
$V_{stall,\,AP}$	105	[knots]	Stall Speed (approach) - (CAS)
$C_{D0,\,AP}$	0.0477	N/A	Parasitic drag coefficient (approach)
$C_{D2,\,AP}$	0.0423	N/A	Induced drag coefficient (approach)
$V_{stall,\,LD}$	103	[knots]	Stall Speed (landing) - (CAS)
$C_{D0,\,LD}$	0.0653	N/A	Parasitic drag coefficient (landing)
$C_{D2,\,LD}$	0.0412	N/A	Induced drag coefficient (landing)
$C_{D0,\,LDG}$	0.0235	N/A	Parasitic drag coefficient (landing gear)

Engine Thrust

Symbols	Value	Units	Description
$C_{Tc,1}$	145730	[N]	1st maximum climb thrust coefficient
$C_{Tc,2}$	55638	[ft]	2nd maximum climb thrust coefficient
$C_{Tc,3}$	0.14200E-10	$[1/ft^2]$	3rd maximum climb thrust coefficient
$C_{Tc,4}$	10.7	[K]	1st thrust temperature coefficient
$C_{Tc,5}$	0.0075	[1/K]	2nd thrust temperature coefficient
$C_{T\ des,\,low}$	0.0552	N/A	Low altitude descent thrust coefficient
$C_{T\ des,\,high}$	0.0746	N/A	High altitude descent thrust coefficient
$H_{p,\,des}$	15000	[ft]	Transition altitude for calculation of

			descent thrust
$C_{T\ \text{des, app}}$	0.15	N/A	Approach thrust coefficient
$C_{T\ \text{des, ld}}$	0.29	N/A	Landing thrust coefficient
$V_{\text{des, ref}}$	280	[knots]	Reference descent speed (CAS)
$M_{\text{des, ref}}$	0.78	N/A	Reference descent Mach number

Fuel Flow

Symbols	Value	Units	Description
$C_{\text{f 1}}$	0.9468	$\left[\text{kg}/\left(\text{min}\cdot\text{kN}\right)\right]$	1st thrust specific fuel consumption coefficients
$C_{\text{f 2}}$	0.1000E+15	[knots]	2nd thrust specific fuel consumption coefficients
$C_{\text{f 3}}$	11.031	[kg/min]	1st descent fuel flow coefficients
$C_{\text{f 4}}$	54252	[ft]	2nd descent fuel flow coefficients
$C_{\text{f cr}}$	0.9737	N/A	Cruise fuel flow correction coefficient

Ground Movement

Symbols	Value	Units	Description
TOL	2500	[m]	Take-off length
LDL	1850	[m]	Landing length
Span	35.8	[m]	Wingspan
Length	33.6	[m]	Length

A.2 Aircraft A2

Aircraft Type

Symbols	Value	Description
n_{eng}	2	Number of engines
Engine type	CFM56_5A3	Jet
Wake Category	M	Medium

Aircraft Mass

Symbols	Value	Units	Description
m_{ref}	64000	[Kg]	Reference mass
m_{min}	39000	[Kg]	Minimum mass
m_{max}	77000	[Kg]	Maximum mass
m_{pyld}	21500	[Kg]	Maximum payload mass

Flight Envelope

Symbols	Value	Units	Description
V_{MO}	350	[knots]	Maximum operating speed - (CAS)
M_{MO}	0.82	N/A	Maximum operating Mach number
h_{MO}	39000	[ft]	Maximum operating altitude
h_{max}	34354	[ft]	Maximum altitude at MTOW and ISA
G_{t}	-130	[ft/K]	Temperature gradient on maximum altitude

Aerodynamics

Symbols	Value	Units	Description
S	122.6	$\left[\text{m}^2\right]$	Reference wing surface area
C_{Lbo}	1.4041	N/A	Buffet onset lift coefficient
k	0.79242	N/A	Buffeting gradient
$V_{stall,\,\text{CR}}$	145	[knots]	Stall Speed (cruise) - (CAS)
$C_{D0,\,\text{CR}}$	0.024	N/A	Parasitic drag coefficient (cruise)
$C_{D2,\,\text{CR}}$	0.0375	N/A	Induced drag coefficient (cruise)
$V_{stall,\,\text{IC}}$	120	[knots]	Stall Speed (initial climb) - (CAS)
$C_{D0,\,\text{IC}}$	0.024	N/A	Parasitic drag coefficient (initial climb)
$C_{D2,\,\text{IC}}$	0.0469	N/A	Induced drag coefficient (initial climb)
$V_{stall,\,\text{TO}}$	114	[knots]	Stall Speed (take-off) - (CAS)
$C_{D0,\,\text{TO}}$	0.039	N/A	Parasitic drag coefficient (Take-off)
$C_{D2,\,\text{TO}}$	0.0396	N/A	Induced drag coefficient (Take-off)
$V_{stall,\,\text{AP}}$	107	[knots]	Stall Speed (approach) - (CAS)
$C_{D0,\,\text{AP}}$	0.0456	N/A	Parasitic drag coefficient (approach)
$C_{D2,\,\text{AP}}$	0.0381	N/A	Induced drag coefficient (approach)

$V_{stall,\,\mathrm{LD}}$	101	[knots]	Stall Speed (landing) - (CAS)
$C_{D0,\,\mathrm{LD}}$	0.0838	N/A	Parasitic drag coefficient (landing)
$C_{D2,\,\mathrm{LD}}$	0.0371	N/A	Induced drag coefficient (landing)
$C_{D0,\,\mathrm{LDG}}$	0.0312	N/A	Parasitic drag coefficient (landing gear)

Engine Thrust

Symbols	Value	Units	Description
$C_{Tc,1}$	136050	[N]	1st maximum climb thrust coefficient
$C_{Tc,2}$	52238	[ft]	2nd maximum climb thrust coefficient
$C_{Tc,3}$	2.6637E-11	$\left[1/\mathrm{ft}^2\right]$	3rd maximum climb thrust coefficient
$C_{Tc,4}$	10.29	[K]	1st thrust temperature coefficient
$C_{Tc,5}$	0.005845	[1/K]	2nd thrust temperature coefficient
$C_{T\ \mathrm{des,\,low}}$	0.09437	N/A	Low altitude descent thrust coefficient
$C_{T\ \mathrm{des,\,high}}$	0.131014	N/A	High altitude descent thrust coefficient
$H_{p,\,\mathrm{des}}$	15000	[ft]	Transition altitude for calculation of descent thrust
$C_{T\ \mathrm{des,\,app}}$	0.13	N/A	Approach thrust coefficient
$C_{T\ \mathrm{des,\,ld}}$	0.34	N/A	Landing thrust coefficient
$V_{\mathrm{des,\,ref}}$	310	[knots]	Reference descent speed (CAS)
$M_{\mathrm{des,\,ref}}$	0.78	N/A	Reference descent Mach number

Fuel Flow

Symbols	Value	Units	Description
$C_{\mathrm{f}\,1}$	0.94	$\left[\mathrm{kg}/\left(\mathrm{min}\cdot\mathrm{kN}\right)\right]$	1st thrust specific fuel consumption coefficients
$C_{\mathrm{f}2}$	100000	[knots]	2nd thrust specific fuel consumption coefficients
$C_{\mathrm{f}3}$	8.89	[kg/min]	1st descent fuel flow coefficients
$C_{\mathrm{f}4}$	81926	[ft]	2nd descent fuel flow coefficients
$C_{\mathrm{f\,cr}}$	1.06	N/A	Cruise fuel flow correction coefficient

Ground Movement

Symbols	Value	Units	Description
TOL	2190	[m]	Take-off length
LDL	1440	[m]	Landing length
Span	34.1	[m]	Wingspan
Length	37.57	[m]	Length

Annex B

B. Relevant Publications

This thesis consists of the following journal publications and conference communications:

Journal Publications

- K. Ahmed, and K. Bousson, "Generating time optimal trajectory from predefined 4D waypoint networks," *International Review of Aerospace Engineering (IREASE)*, vol. 10, no. 4, pp. 207-214, 2017. DOI: 10.15866/irease.v10i4.12570.

- K. Ahmed, K. Bousson, and M. Coelho, "Spline parameterization based nonlinear trajectory optimization along 4D waypoints," *Advances in Aircraft and Spacecraft Science*, vol. 6, no.5, pp. 391-407, 2019. DOI: 10.12989/aas.2019.6.5.391.

- K. Ahmed, M. Coelho, and K. Bousson, "Optimal fuel saving in 4D waypoint networks," *International Congress on Engineering — Engineering for Evolution,* KnE Engineering, pp. 500–513, 2020. DOI 10.18502/keg.v5i6.7065.

- K. Ahmed, K. Bousson, and M. Coelho, "A modified dynamic programming approach for 4D minimum fuel and emissions trajectory optimization," *Aerospace*, vol. 8, no. 5: 135, 2021. DOI: 10.3390/aerospace8050135.

Conference Communications

- K. Ahmed, "4D commercial trajectory optimization for fuel saving and environmental impact reduction," *2nd Doctoral Congress of Engineering* (DCE17), Porto, Portugal, June 2017.

- K. Ahmed, K. Bousson, and M. Coelho, "Blended cubic spline based nonlinear trajectory optimization," *4EJIL (4º Encontro de Jovens Investigadores do LAETA)*, Covilhã, Portugal, November 2017.

- K. Ahmed, and M. Coelho, "Nonlinear trajectory optimization along 4D waypoints based on spline parameterization," *International Congress on Engineering (ICEUBI)*, Covilhã, Portugal, December 2017.

- K. Ahmed, K. Bousson, and M. Coelho, "Optimal Fuel Saving in 4D Waypoint Networks," *International Congress on Engineering (ICEUBI)*, , Covilhã, Portugal, November 2019.

- K. Ahmed, K. Bousson, and M. Coelho, "4D flight trajectory optimization by modified dynamic programming," *Aerospace Europe Conference 2020 (AEC2020)*, Bordeaux, France, February 2020.