An Optimal Control Approach to Flight Management Systems for Unmanned Aerial Vehicles

Michael Di Perna

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical Engineering) at

Concordia University

Montréal, Québec, Canada

April 2017

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Michael Di Perna**

Entitled: **An Optimal Control Approach to Flight Management Systems for Unmanned Aerial Vehicles**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

__________________________________ Chair
Dr. Rabin Raut

__________________________________ External Examiner
Dr. Walter Lucia

__________________________________ Examiner
Dr. Amir Aghdam

__________________________________ Supervisor
Dr. Luis Rodrigues

Approved by __________________________________
William E. Lynch, Chair
Department of Electrical and Computer Engineering

____________ 2017

____
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

An Optimal Control Approach to Flight Management Systems for Unmanned Aerial Vehicles

Michael Di Perna

With unmanned aerial vehicles (UAVs) becoming increasingly present in military and commercial applications, the flight path efficiency and integration with current manned aircraft become important research topics to address in the coming years. This thesis considers three problems relating to UAVs: the optimal control of a single quadrotor UAV, a multi-agent coverage problem, and a software flight management system which can be used on UAVs.

The optimal control problem for a quadrotor UAV is considered with a tuning parameter, the cost index, used in flight management systems to trade-off between time and energy costs. A state-feedback control law is developed and simulation results are presented. A software flight management system (SFMS) using aerospace standard communication protocols is developed and validated with an industry flight simulator. The SFMS allows for the testing of algorithms which can be used on real aircraft (manned or unmanned) without requiring access to a costly commercial flight management system. An energy efficient coverage problem from previous work is considered and extended to include agents with second order dynamics using the backstepping technique. The extension to second order dynamics requires the analysis of the dynamics of Voronoi cells. A geometric interpretation is presented for the change in area and change in position of the center of mass for Voronoi cells. Simulation results are presented comparing the first order and second order agents.

Acknowledgments

I would like to direct a special acknowledgement to my supervisor Dr. Luis Rodrigues, who has provided me with not only the tools and guidance to complete this thesis, but also with many opportunities to work with industry. I thank him for his continuous patience and dedication towards myself as a person and my work.

Thank you to all my colleagues in the Hybrid Control Systems (HYCONS) lab for their support: Michael El-Jiz, Jesus, Miad, Manuel, Alex, Maxim, Emily, and Reza.

A special thanks to my family and girlfriend for their support and encouragement throughout my degree.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

One of the greatest challenges facing humanity in the upcoming century is global warming. This is the conclusion of a study that was done by the World Health Organization in [6]. It is generally agreed that emissions from burning fossil fuels have had a large impact on global warming. The aviation industry has been steadily growing and has reached over 100,000 daily commercial flights [7]. Aircraft are estimated to account for 2% of the global greenhouse emissions, and the amount of carbon emissions is expected to grow by 3-4 % per year [8]. Meanwhile, UAVs (unmanned aerial vehicles) is a growing industry [9] and can be found in a variety of different applications such as package delivery (Amazon), surveillance for rescue, security, and defense, and providing internet coverage (Google and Facebook). NASA has also been investigating autonomous aircraft for cargo delivery and perhaps even passenger aircraft. Although many UAVs utilize alternate energy sources, such as hydrogen cells and batteries, their efficiency will ultimately determine their usefulness and their cost-effectiveness. To increase the flight time and cost-effectiveness of UAVs one may either improve the powerplant (i.e. higher energy density batteries) or one may reduce the amount of fuel or energy consumed during flight. Current commercial aircraft utilize a measure called the cost index, C_I, which denotes a trade-off between time and fuel. The speed of the aircraft is dictated by the value of C_I chosen before the flight. The cost index allows airlines to determine the optimal trajectories of their aircraft to reduce overall cost of the flight (i.e. time related costs and fuel cost).

The concept of using a cost index for cost-effective flights will be extended to UAVs in this thesis. This thesis will also address a multi-agent system by solving a coverage problem with considerations for energy-efficient deployment of agents with second order dynamics. The multi-agent result can then be used in the aforementioned applications, such as search and rescue.

1.2 Objectives

The objectives of this thesis are as follows:

- To develop an analytic state-feedback control law for the optimal control of UAVs in the context of a flight management system.
- To develop a state-feedback control law for agents with second order dynamics in a multi-agent coverage problem using the dynamics of Voronoi cells.
- To develop a software flight management system as a research platform for testing and validating future research.

1.3 Literature Review

1.3.1 Optimal Control of Quadrotors

The field of optimal control provides tools which allow engineers to perform optimization on the control of dynamic systems. We are specifically interested in generating control laws for quadrotors which optimize its energy consumed while still providing sufficient performance.

The work in [10] presents an LQT (linear quadratic tracking) controller for quadrotors. They assume a fixed final time with an unconstrained final state, this leads to time-varying gains that are computed offline. The authors in [11] propose a novel backstepping technique for the optimal control of a nonlinear quadrotor model. This technique requires the cost function to have an infinite horizon, and does not allow the direct use of a cost index in the cost function. Reference [12] develops a detailed model of the quadrotor and applies an LQR controller after linearizing the dynamics about the operating point. Numerical methods for solving optimal control problems for quadrotors are considered in [13]. It was shown that the methods were effective at producing trajectories for obstacle avoidance and aggressive maneuvers. The authors claim that it may take a few seconds to generate a trajectory while using an Intel Xeon processor at 3.4 GHz. Although this would be acceptable for offline calculations, we suspect that in real-time the hardware aboard a modestly priced quadrotor (or low cost quadrotor) would struggle to generate real-time trajectories in a reasonable amount of time. The work in [14] presents a model for the DC motors and battery charge to determine minimum energy paths for a quadrotor. The authors also include a formulation which includes a tuning parameter to apply a trade-off between time and energy. The optimal control problems are solved numerically and can take over one minute to compute, which is suited for offline calculation. Reference [15] provides a higher level approach to energy management and looks at predicting the battery life for mission planning. The authors employ a multi-model predictive controller (MMPC) which switches among a bank of models. A particle filter was used to predict the end-of-discharge time for the battery. The work presented in [16] looks at the time dependent optimal trajectories of a quadrotor by using Pontryagin's Maximum Principle. The authors present the findings in the context of a benchmark against which controllers can be compared. The solutions found are computed numerically using a boundary value problem (BVP) solver in Matlab. Reference [17] focuses on the trajectory generation and guidance of a quadrotor for implementation in an UAV FMS. The authors consider the nonlinear dynamics of a quadrotor and a cost function which penalizes the length of the trajectory. A quasi-optimal trajectory is found using the nonlinear programming optimization ”fmincon”. Backstepping is used to synthesize control laws for the quadrotor's attitude and position. Similarly, the work in [18] describes a trajectory generation technique for quadrotors which minimizes a cost function that penalizes the square of angular and translational velocities. The trajectory is solved numerically using ”fmincon”. The authors also describe a collision detection algorithm for trajectory generation. The authors of [19] consider UAV trajectory planning for obstacle avoidance for shared airspace. Mixed integer linear programming (MILP) is employed to numerically solve the trajectory optimization problem. The authors consider fuel consumption, energy, and a penalty on the final time in the cost function. Reference [20] describes a numerical method to produce minimum time trajectories for quadrotors while considering constraints on the quadrotor's configuration, actuator velocities and torques, and obstacle avoidance. Nonlinear dynamics are considered for the quadrotor model but the trajectory generation technique may not be suitable for on-board use as it requires a numerical solution. The work in [21] describes a computationally efficient algorithm for trajectory generation of quadrotors. The paper considers generating trajectories of a quadrotor with a given final time using Pontryagin's Maximum Principle and considers the feasibility of each trajectory generated. The algorithm proposed allows for the computation of many trajectories to meet the physical constraints of the quadrotor (i.e. thrust limit). The trajectories generated are a function of time and a motion capture system is used to provide the closed-loop trajectory tracking.

1.3.2 Flight Management Systems for UAVs

One can find a number of simulated FMSs for consumer flight simulators (i.e. Microsoft Flight Simulator and X-Plane) which try to replicate the interface and functionality of currently available flight management systems. One can take this further by building a flight management system in software which can interface with real aircraft and UAVs using industry standard protocols.

The survey performed in [22] anaylzes the capabilities of available low-cost simulated FMS plugins for Microsoft Flight Simulator (MSFS), in terms of their research potential, in comparison to a General Electric FMS. Although some discrepancies were found in the continous descent arrival (CDA) ground speeds (which were estimated for the MSFS plugins), the lateral and vertical profiles were found to be within acceptable bounds. The use of an external device in the cockpit has been examined in [23] to introduce a low-cost flight data recorder which also has the capability to estimate some aircraft states and provide reports from flights. In contrast to the work presented in this chapter, the external device does not provide any interactions with the aircraft's avionics. In [24], a design methodology for UAV system design is proposed and a simulated example is used to illustrate the procedure. The paper considers the safety-critical components and verification that would be required for use on UAVs. Using more recent technology, [25] proposes an FMS for UAVs which includes intent sharing and communication with ground systems. The intention of the research is to provide the basis for UAVs to function within current aircraft airspace. The authors look at pre-emptively integrating safety measures which may be required for UAV flights in the future. The Sparrow flight management system [26] is a commercially available cloud based web tool FMS for fleets of UAVs which provides scheduling and tracking among other features. Sparrow focuses on flight management in terms of tracking and collecting analytics of UAV(s) flights. This macro approach to flight management system provides tools for studying the costs incurred by flights, but does not allow for the usage of a cost index for a particular flight. The certification and verification of software for UAVs is studied in [27] through the ARTIS project. The paper addresses the need to rethink software development for UAVs when attempting to meet standards for aircraft software certification, such as the standards from the Radio Technical Commission for Aeronautics. The lack of a pilot present in a UAV requires more safety critical code and generally increases the complexity of the code. An FMS performs calculations to provide optimal trajectories dependent on a parameter entered by the pilot called the cost index (C_I). There has been recent research on the cost index to provide analytic solutions for the optimal target airspeed during cruise, climb, and descent [28]. Improvements to the FMS can allow for more fuel efficient flights with the current aircraft in use, but most FMSs are a black-box device which researchers cannot easily access. With the introduction of aircraft interface devices (AIDs) it is possible to connect a tablet or laptop to an aircraft's systems, allowing the development of an external flight management system.

1.3.3 Multi-agent Coverage Problems

Voronoi diagrams can be found in many different fields [29] such as astronomy, geography, chemistry, physics, marketing, and engineering. We can find applications for Voronoi diagrams in engineering for topics such as fluid dynamics, image processing, and coverage problems in multi-agent systems. It is of particular interest in coverage problems to analyze the dynamics of Voronoi cells as generators move because the optimal coverage is obtained when generators lie at the center of mass of their respective Voronoi cell.

The change in volume, which in 2D corresponds to area, of a Voronoi cell due to moving generators is determined by Espanol et al. in [30]. Du et al. [31] use a geometric property, the perpendicular bisector property of Voronoi diagrams, to develop a result which can be used to solve the boundary integral of a Voronoi cell. Cortes et al. [32] solve the boundary integral of a Voronoi cell for scalar functions in the context of a coverage problem. A time-varying density function for a coverage problem is considered by Diaz-Mercade et al. in [33]. They also obtain an expression for the change in the position of the center of mass with respect to agents, but it is left in an integral form and is not developed any further. The work in [34] formulates the multi-agent coverage problem as an optimal control problem and looks to providing an energy efficient control scheme by introducing tuning parameters in the optimization.

The work in [35] describes a coverage problem with the usage of Power Diagrams. Each agent is assigned a weight that describes its coverage range. The agents have first order dynamics and the weight variable is also considered to have first order dynamics. Each region in the partition is assigned a workload which is application dependent and can be described by a stochastic function if desired. A continuation of this work is presented in [36] where equitable partitions (based on a measure function) of a region are computed using Power Diagrams. The results are then used in application examples such as dynamic vehicle routing for mobile robotic networks and wireless ad hoc networks. Reference [37] describes a coverage problem involving agents which have limited-range communications and anisotropic sensing (directional sensors). The solution presented uses an approximation to achieve a distributed solution to the problem proposed. Both the agents and their sensors have first order dynamics.

1.4 Contributions

The contributions of this thesis are as follows:

- An optimal control problem is solved for a quadrotor UAV flight using a cost index to trade-off between energy and time. A state-feedback law is obtained for the quadrotor's flight in 2D, the trade off between control effort and time is presented in a Pareto trade-off curve, and the maximum velocity of the quadrotor can be used to determine a range of allowable cost indices.

- A distributed control law for a multi-agent coverage problem is obtained using the backstepping technique. Geometric derivations for the change of mass and change of center of mass for Voronoi cells are found to enable the implementation of the control law.

- An implementation of a FMS in software is developed using aerospace communication standards and validated using an industry flight simulator.

1.5 Structure of the Thesis

Chapter 2 begins by providing the preliminaries for quadrotor dynamics and optimal control. This is followed by a time-dependent control law and then a state-feedback law for quadrotors which includes a cost index term as a tuning parameter. Simulation results are presented for both time-dependent and state-feedback control schemes and a Pareto trade-off curve is illustrated for the state-feedback simulation. Path following for fixed-wing UAVs in the lateral plane is also discussed and a control law is developed. Next, Chapter 3 covers preliminaries on aircraft dynamics, spherical trigonometry for aircraft navigation, and the optimal cruise speed for aircraft. These topics are then used in a software flight management system framework for UAVs. The design and functionality of the software FMS is presented along with experimental results obtained by interfacing the software FMS with an industry flight simulator. Chapter 4 covers Voronoi diagrams and Reynolds Transport Theorem and applies these concepts to an optimal control multi-agent coverage problem with agents possessing second order dynamics. The change in area and the change in the position of the center of mass of Voronoi cells are considered when developing a control law for the agents. Chapter 5 concludes the work presented in the thesis.

Chapter 2

Optimal Control Framework for UAV Flight Management Systems

This chapter will consider the control of a quadrotor UAV which is determined by providing a cost index, as found in aircraft flight management systems. We will also consider the trajectory generation for the lateral movement of a fixed-wing UAV. Although a flight management system is a very complex system (see chapter 3), we will consider the performance function of the FMS as it relates to energy and cost effectiveness.

In this chapter we will present a time-dependent control law that provides a reference trajectory when considering both the kinetic and potential energy of the quadrotor. We will also present a state-feedback control law which is obtained analytically and does not require an infinite horizon. The state-feedback controller allows the determination of a range of tuning parameters based on the maximum velocity of the quadrotor.

2.1 Preliminaries

2.1.1 Quadrotor Mathematical Model

A quadrotor is an aerial vehicle which rotates by producing differentials in thrust between it's four motors. The following section presents the mathematical model of the quadrotor.

Reference Frames

An inertial reference frame will be considered attached to the center of a tangent plane on the surface of the Earth [1]. For a local area around the center of the tangent plane, the Earth can be considered flat. The inertial coordinates are x, y, and z, where $e_x \times e_y = e_z$. Let e_x be the unit vector in the direction of the north pole and e_z be the upwards unit normal vector to the tangent plane.

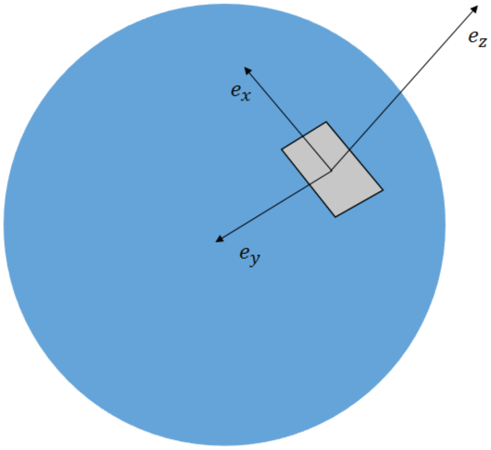


Figure 2.1: Reference frame attached to a tangent plane on Earth.

The quadrotor's orientation, between the body frame B and a reference frame F, can be represented through three rotations using Z-Y-X Euler angles roll, pitch, and yaw. The body axes are fixed to the center of the quadrotor's body and are illustrated in Fig. 2.2. The orientation of the body frame B relative to the reference frame F is

$$ {}^{F}_{B}R(\phi, \theta, \psi) = R_z(\psi) R_y(\theta) R_x(\phi) \tag{1}$$

[1] Although the Earth is not an inertial reference frame due to its rotation and movement, we can accurately approximate it as an inertial reference frame for the applications considered in this thesis due to the relatively small magnitude of the accelerations experienced by the Earth

where

$$
\begin{aligned}
R_x(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\
R_y(\theta) &= \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \\
R_z(\psi) &= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{2}
$$

and ϕ is the roll angle, θ is the pitch angle, and ψ is the yaw angle.

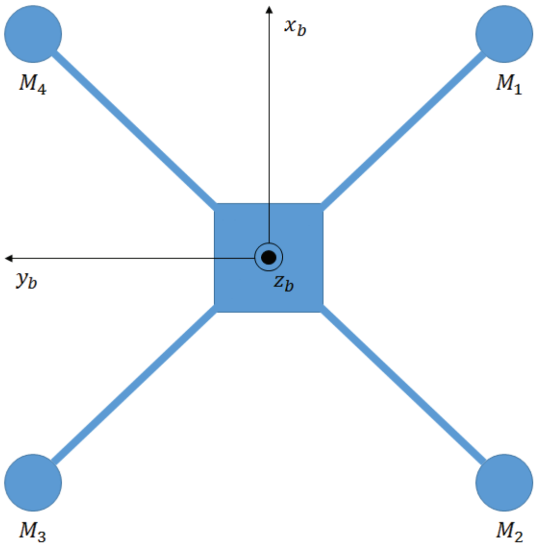


Figure 2.2: Quadrotor body axes and motors.

Quadrotor Dynamics

The following section presents the quadrotor's dynamics using Newton-Euler's equations. We will consider the following assumptions:

Assumption 1. *The quadrotor is a rigid body.*

Assumption 2. *The quadrotor flies at low speeds such that drag can be neglected.*

Assumption 3. *The quadrotor is symmetrical relative to the longitudinal x-z plane.*

Assumption 4. *The quadrotor will operate near hover conditions.*

The quadrotor has four motors which generate forces along the body's z_b direction. Each motor generates a force proportional to the square of the propeller's angular velocity. The force can be expressed as follows [38]:

$$F_i = k_f \omega_i^2 \tag{3}$$

where i denotes the index of the motor, k_f is the thrust coefficient of the propeller, and ω_i is the angular velocity of motor i. Let us define $\mathbf{U} = [U_1, U_2, U_3, U_4]^T$ to be the control input vector which describes the force U_1 along the z_b axis and the torques U_2, U_3, and U_4 around x_b, y_b, and z_b respectively. The control input force and torques are [39]:

$$\begin{aligned}
U_1 &= F_1 + F_2 + F_3 + F_4 \\
U_2 &= l(F_3 + F_4 - F_1 - F_2) \\
U_3 &= l(F_1 + F_4 - F_2 - F_3) \\
U_4 &= c_\tau(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2)
\end{aligned} \tag{4}$$

where l is the distance from the center of mass of the quadrotor to the center of a motor and c_τ is the reaction torque coefficient which can be experimentally determined. The translational dynamics in the inertial frame are [40]:

$$\begin{aligned}
\ddot{x} &= (\sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi))\frac{U_1}{m} \\
\ddot{y} &= (\sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi))\frac{U_1}{m} \\
\ddot{z} &= -g + (\cos(\theta)\cos(\phi))\frac{U_1}{m}
\end{aligned} \tag{5}$$

where m is the mass of the quadrotor in kilograms and g is the acceleration due to gravity ($9.81 m/s^2$). The translational dynamics of the quadrotor in the longitudinal plane, for constant height, can be derived from equation (5) by letting $\phi = 0$, $\psi = 0$, and $\ddot{z} = 0$, and is expressed as follows:

$$\begin{aligned} mg &= T\cos\theta \\ m\dot{v} &= T\sin\theta \end{aligned} \tag{6}$$

where T is the total thrust generated by the quadrotor (equivalent to U_1). The rotational dynamics are described by [41]:

$$\begin{aligned} \ddot{\phi} &= \frac{I_{yy} - I_{zz}}{I_{xx}}\dot{\theta}\dot{\psi} - \frac{J_{TP}}{I_{xx}}\dot{\theta}\Omega + \frac{U_2}{I_{xx}} \\ \ddot{\theta} &= \frac{I_{zz} - I_{xx}}{I_{yy}}\dot{\phi}\dot{\psi} + \frac{J_{TP}}{I_{yy}}\dot{\phi}\Omega + \frac{U_3}{I_{yy}} \\ \ddot{\psi} &= \frac{I_{xx} - I_{yy}}{I_{zz}}\dot{\phi}\dot{\theta} + \frac{U_4}{I_{zz}} \\ \Omega &= -\omega_1 + \omega_2 - \omega_3 + \omega_4 \end{aligned} \tag{7}$$

where J_{TP} is the rotational moment of inertia about the propeller axis. In equation (7), we have used assumption 3 leading to the simplification $I_{xy} = I_{yz} = 0$. Since I_{xz} is much smaller than I_{xx}, I_{yy} and I_{zz}, it can be neglected. Assumption 4 allows the following approximations to be made: $\phi \approx 0$, $\theta \approx 0$, and $\psi \approx \psi_0$, resulting in the following simplified translational dynamics:

$$\begin{aligned} \ddot{x} &= (\sin(\psi_0)\phi + \cos(\psi_0)\theta)\frac{U_1}{m} \\ \ddot{y} &= (\sin(\psi_0)\theta - \cos(\psi_0)\phi)\frac{U_1}{m} \\ \ddot{z} &= -g + \frac{U_1}{m} \end{aligned} \tag{8}$$

where ψ_0 is a constant yaw angle. Linearizing equation (7) about the hovering state, with $\dot{\phi} \approx 0$ and $\dot{\theta} \approx 0$ yields the following attitude dynamics

$$\begin{aligned} \ddot{\phi} &= \frac{U_2}{I_{xx}} \\ \ddot{\theta} &= \frac{U_3}{I_{yy}} \\ \ddot{\psi} &= \frac{U_4}{I_{zz}} \end{aligned} \tag{9}$$

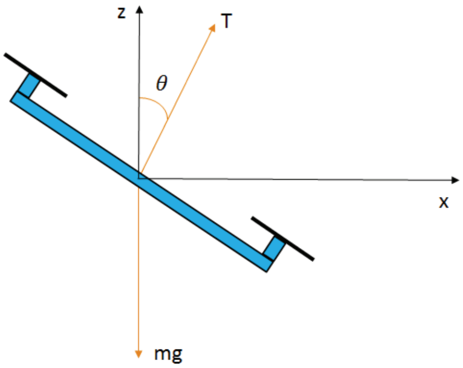


Figure 2.3: Longitudinal axes of a quadrotor.

2.1.2 Review of Optimal Control

There are many instances when a control engineer must face the trade-off between performance and control effort when designing a controller for a system. The optimal control framework provides a tool for engineers to weight parameters representing the performance of the system and determine the ”optimal” control input to provide the best performance. In it’s simplest form, we can define a function $L(\mathbf{u})$ that we wish to minimize by selecting the appropriate control input vector $\mathbf{u} = [u_1, u_2, ..., u_m]^T$[42][43]. Assuming there are no constraints on the control input vector and the first and second partial derivatives of $L(\mathbf{u})$ exist, the necessary conditions for a minimum are

$$\frac{\partial L(\mathbf{u})}{\partial u_i} = 0,\ i = 1, ..., m \tag{10}$$

and

$$\frac{\partial^2 L(\mathbf{u})}{\partial \mathbf{u}^2} \geq 0 \tag{11}$$

Equation (11) is verified when the eigenvalues of the Hessian are greater than or equal to zero, where the Hessian matrix is defined as

$$\mathcal{H} = \frac{\partial^2 L(\mathbf{u})}{\partial \mathbf{u}^2} = \begin{bmatrix} \frac{\partial^2 L(\mathbf{u})}{\partial u_1^2} & \frac{\partial^2 L(\mathbf{u})}{\partial u_1 \partial u_2} & \cdots & \frac{\partial^2 L(\mathbf{u})}{\partial u_1 \partial u_m} \\ \frac{\partial^2 L(\mathbf{u})}{\partial u_2 \partial u_1} & \frac{\partial^2 L(\mathbf{u})}{\partial u_2^2} & \cdots & \frac{\partial^2 L(\mathbf{u})}{\partial u_2 \partial u_m} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 L(\mathbf{u})}{\partial u_m \partial u_1} & \frac{\partial^2 L(\mathbf{u})}{\partial u_m \partial u_2} & \cdots & \frac{\partial^2 L(\mathbf{u})}{\partial u_m^2} \end{bmatrix} \tag{12}$$

The sufficient condition for a minimum is satisfied when the Hessian matrix is positive definite (strictly positive eigenvalues)

$$\frac{\partial^2 L(\mathbf{u})}{\partial \mathbf{u}^2} > 0 \tag{13}$$

In engineering applications, we are often interested in dynamic systems, i.e. systems that have internal states and differential equations which govern the evolution of the those states. Let **x** represent the vector of states for a given system. The dynamics of the system are expressed as follows:

$$\dot{x}_i = f_i(\mathbf{x}, \mathbf{u}, t) \tag{14}$$

where $i = 1...N$, N is the number of states in the state vector $\mathbf{x} = [x_1, x_2...x_N]^T$, **u** is the vector of control inputs, and t is time. Let us now consider the following scalar valued cost function

$$J = \Phi(\mathbf{x}(t_f), t_f) + \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)\, \mathrm{d}t \tag{15}$$

where $L(\mathbf{x}(t), \mathbf{u}(t), t)$ is known as the running cost and $\Phi(\mathbf{x}(t_f), t_f)$ is a scalar valued penalty on the final state and final time t_f. Let us consider the final time t_f to be unspecified. We may also consider that the states of the system have initial values $\mathbf{x}_0$ and some specified final values denoted by the terminal constraint function $\Psi(\mathbf{x}(t_f), t_f) = 0$. The optimal control problem can then be formulated as follows:

$$\begin{aligned} J^* = \inf_{\mathbf{u}(t)} &\left(\Phi(\mathbf{x}(t_f), t_f) + \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)\,\mathrm{d}t \right) \\ &s.t. \\ &\dot{x}_i(t) = f_i(\mathbf{x}, \mathbf{u}, t),\ i = 1...N \\ &\mathbf{x}(0) = \mathbf{x}_0 \\ \Psi(\mathbf{x}(t_f), t_f) &= 0 \end{aligned} \tag{16}$$

where J^* denotes the optimal cost.

Pontryagin's Maximum Principle (PMP)

Pontryagin's Maximum principle provides the necessary conditions to find the optimal control law $\mathbf{u}^*(t)$ to minimize the cost function in equation (16). We can append the terminal constraints and dynamics to the cost function as follows

$$J = (\Phi + \nu^T \Psi)_{t=t_f} + \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t) + \sum_{i=1}^{N} \lambda_i^T (f_i - \dot{x}_i)\,\mathrm{d}t \tag{17}$$

where λ_i is a Lagrange multiplier for the dynamic constraint $\dot{x}_i = f_i$ and ν is a vector of Lagrange multipliers for the terminal constraints in Ψ. The Hamiltonian is defined as

$$H = L(\mathbf{x}(t), \mathbf{u}(t), t) + \sum_{i=1}^{N} \lambda_i^T f_i \tag{18}$$

Using variational calculus for the case when the final time is unspecified, the following equations are obtained as necessary conditions for optimality [43]

$$\frac{\partial H}{\partial \mathbf{u}} = \mathbf{0} \tag{19a}$$

$$\dot{\lambda}_i = -\frac{\partial H}{\partial x_i} \tag{19b}$$

$$\lambda^T(t_f) = (\Phi_x + \nu^T \Psi_x)_{t=t_f} \tag{19c}$$

$$\Omega = (\Phi_t + \nu^T \Psi_t + (\Phi_x + \nu^T \Psi_x)\mathbf{f} + L)_{t=t_f} = 0 \tag{19d}$$

The Lagrange multiplier λ_i will henceforth be referred to as the co-state to state x_i. For the case of free final time and if the running cost L is not an explicit function of time, the Hamiltonian along the optimal trajectory is zero [42]:

$$H^* = 0 \tag{20}$$

2.2 Optimal Trajectory of a Single Quadrotor

2.2.1 Time-dependent Optimal Control

Let us consider a quadrotor moving along a single axis while maintaining a constant height (see Figure 2.3). Using (6) the quadrotor can be modeled as a nonlinear, second order system:

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= g\, tan(\theta) \end{aligned} \tag{21}$$

where x is the quadrotor's position, v is it's velocity, θ is the pitch angle, and g is the acceleration due to gravity. The pitch angle will be considered as the control input. Let us assume that the quadrotor will operate near hover conditions, which allows the following approximation to be used: $tan(\theta) \approx \theta$. The linearized dynamics are

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= g\theta \end{aligned} \tag{22}$$

The specified initial and final conditions are as follows

$$\begin{aligned} x(0) &= x_0 \\ v(0) &= v_0 \\ x(t_f) &= x_f \\ v(t_f) &= v_f \end{aligned} \tag{23}$$

Consider the following cost function to be minimized:

$$J = \int_0^{t_f} \frac{1}{2}\alpha v^2 + \frac{1}{2} r\theta^2 + C_I \,\mathrm{d}t \tag{24}$$

where α is the weighting parameter on the velocity of the quadrotor, C_I is the cost index (a penalty on the final time), and r is a weighting parameter on the square of the control input. If we select $\alpha = m$ and $C_I = mgh_0$ then the cost function in (24) considers the kinetic energy, potential energy, and the square of the control input. Although we are considering constant height, the term mgh_0 can be seen as a penalty on the cruise height of the quadrotor. A larger height results in a higher time penalty. This was chosen to provide consideration for the time it may take for the quadrotor to ascend and descend to its target height h_0 without incorporating it directly into the problem. Substituting the cost function and dynamics into equation (18) yields the following Hamiltonian

$$H = \frac{1}{2}\alpha v^2 + \frac{1}{2} r\theta^2 + \lambda_1 v + \lambda_2 g\theta + C_I \tag{25}$$

Taking the partial derivative of the Hamiltonian with respect to the control input yields the following necessary condition for optimality (10):

$$\frac{\partial H}{\partial \theta} = \lambda_2 g + r\theta = 0 \tag{26}$$

Solving equation (26) for θ:

$$\theta^* = -\frac{g}{r}\lambda_2 \tag{27}$$

where θ^* represents the optimal control input. We also obtain the following property of r from the sufficient condition (13):

$$\frac{\partial^2 H}{\partial \theta^2} = r > 0 \tag{28}$$

The time derivatives of the co-states from (19b) are as follows:

$$\begin{aligned} \dot{\lambda}_1 &= -\frac{\partial H}{\partial x} = 0 \\ \dot{\lambda}_2 &= -\frac{\partial H}{\partial v} = -\alpha v - \lambda_1 \end{aligned} \tag{29}$$

From equation (29) we can note that λ_1 does not vary as a function of time. Taking the second time derivative of λ_2 in (29) yields

$$\ddot{\lambda}_2 = -\alpha\dot{v} = -\alpha g\theta \tag{30}$$

Substituting the optimal control input from (27) into equation (30) yields

$$\ddot{\lambda}_2 = -g^2\frac{\alpha}{r}\lambda_2 \tag{31}$$

The solution of the second order differential equation is of the form:

$$\lambda_2(t) = Ae^{g\sqrt{\frac{\alpha}{r}}(t_f-t)} + Be^{-g\sqrt{\frac{\alpha}{r}}(t_f-t)} \tag{32}$$

where A and B are constants. Let us consider the linearized dynamics

$$\begin{aligned}\dot{x} &= v\\ \dot{v} &= g\theta = -\frac{g^2}{r}\lambda_2\end{aligned} \tag{33}$$

and substitute equation (32) for λ_2 yielding

$$\dot{v} = -\frac{g^2}{r}\left(Ae^{g\sqrt{\frac{\alpha}{r}}(t_f-t)} + Be^{-g\sqrt{\frac{\alpha}{r}}(t_f-t)}\right) \tag{34}$$

Integrating equation (34) yields the following expressions for the position and velocity of the quadrotor

$$\begin{aligned}v(t) &= \frac{g}{\sqrt{r\alpha}}Ae^{g\sqrt{\frac{\alpha}{r}}(t_f-t)} - \frac{g}{\sqrt{r\alpha}}Be^{-g\sqrt{\frac{\alpha}{r}}(t_f-t)} + C\\ x(t) &= -\frac{1}{\alpha}Ae^{g\sqrt{\frac{\alpha}{r}}(t_f-t)} - \frac{1}{\alpha}Be^{-g\sqrt{\frac{\alpha}{r}}(t_f-t)} + Ct + D\end{aligned} \tag{35}$$

Using the initial and final conditions, the following five equations are obtained

$$x(0) = -\frac{1}{\alpha}Ae^{g\sqrt{\frac{\alpha}{r}}t_f} - \frac{1}{\alpha}Be^{-g\sqrt{\frac{\alpha}{r}}t_f} + D = x_0 \tag{36a}$$

$$v(0) = \frac{g}{\sqrt{r\alpha}}Ae^{g\sqrt{\frac{\alpha}{r}}t_f} - \frac{g}{\sqrt{r\alpha}}Be^{-g\sqrt{\frac{\alpha}{r}}t_f} + C = v_0 \tag{36b}$$

$$x(t_f) = -\frac{1}{\alpha}A - \frac{1}{\alpha}B + Ct_f + D = x_f \tag{36c}$$

$$v(t_f) = \frac{g}{\sqrt{r\alpha}}A - \frac{g}{\sqrt{r\alpha}}B + C = v_f \tag{36d}$$

$$0 = \frac{1}{2}\alpha v_f^2 + C_I - \frac{g^2}{2r}(A+B)^2 + \lambda_1 v_f \tag{36e}$$

where (36e) is obtained from (19d). The constants (A, B, C, and D) and t_f can be solved numerically. One can also specify a final time and obtain an analytic solution for the constants. The optimal control law is then

$$\theta^*(t) = -\frac{g}{r}\left(Ae^{g\sqrt{\frac{m}{r}}(t_f-t)} + Be^{-g\sqrt{\frac{m}{r}}(t_f-t)}\right) \tag{37}$$

Let us consider a flight with $C_I = mgh_0$, $h_0 = 1$, $m = 1$, $\alpha = m$, $r = 100$, $v_0 = v_f = 0$, $x_0 = -100$, and $x_f = 0$. Solving (36) yields the following values:

$$\begin{aligned} A &= -1.471193373 \cdot 10^{-10} \\ B &= 4.515236410 \\ C &= 4.429446919 \\ D &= -104.5152364 \\ t_f &= 24.61491803 \end{aligned} \tag{38}$$

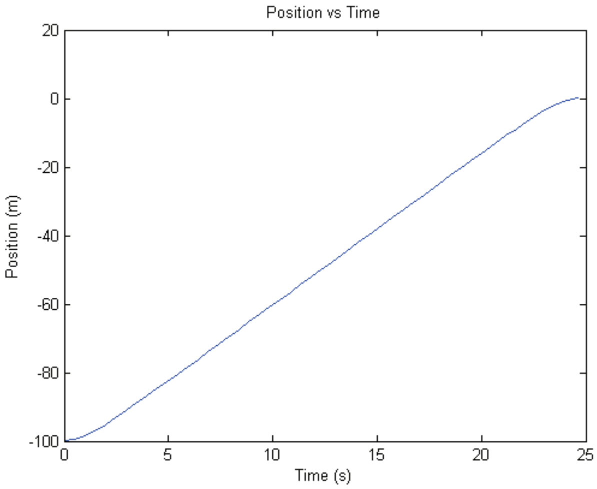


Figure 2.4: Position of the quadrotor as a function of time.

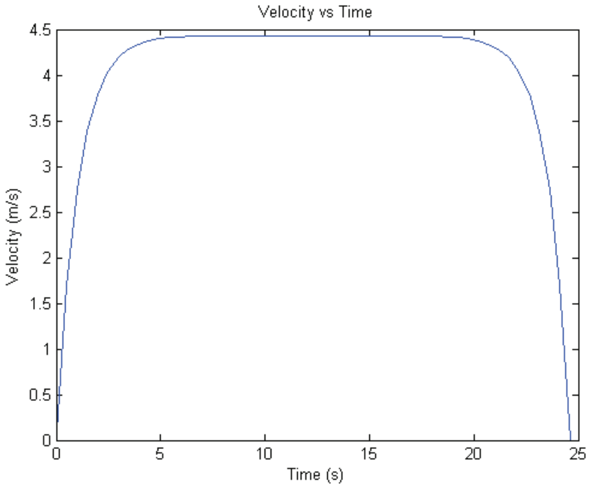


Figure 2.5: Velocity of the quadrotor as a function of time.

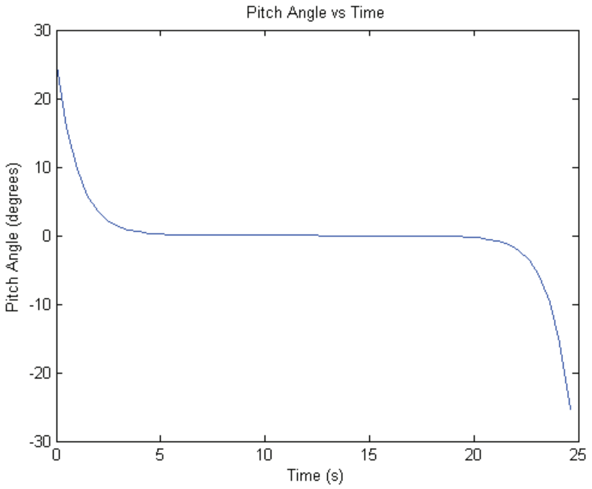


Figure 2.6: Pitch angle of the quadrotor as a function of time.

Figures 2.4, 2.5, and 2.6 illustrate the optimal trajectory the quadrotor should follow if the dynamics were indeed linear. The position and velocity of the quadrotor at the final time are zero as desired. The pitch angle of the quadrotor is sufficiently small for the small angle approximation to apply, although one could use the tuning parameters to trade-off a smaller pitch angle for a longer final time. Applying the optimal control law in (37) to a quadrotor model with nonlinear dynamics, equation (21), results in the trajectories shown in Figures 2.7 and 2.8. Most notably, one can see that the position of the quadrotor does not end at zero as desired due to the error caused by linearization and the control law is time-dependent or open-loop instead of a more robust state-feedback law. This control law would not be suitable for implementation in a real system. The next section will address this issue.

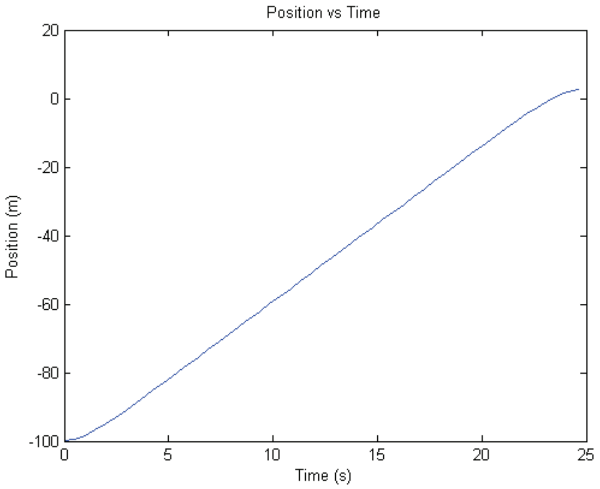


Figure 2.7: Position of the quadrotor as a function of time.

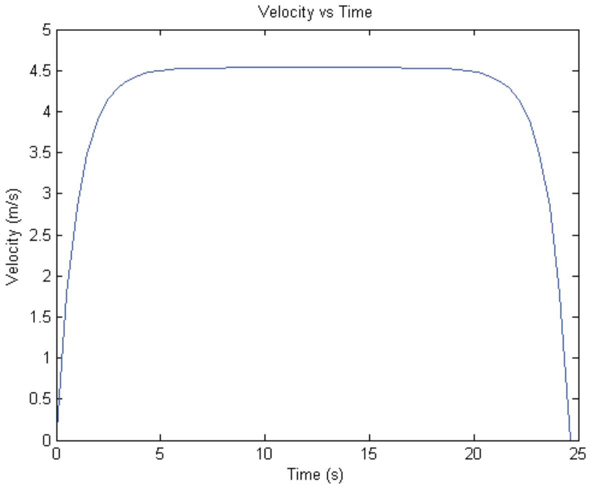


Figure 2.8: Velocity of the quadrotor as a function of time.

2.2.2 State-feedback Optimal Control

Let us again consider the problem of a quadrotor moving from one point to another, at a constant height. We will consider the longitudinal dynamics of the quadrotor as illustrated in Fig. 2.3 and will consider that the final time to reach the desired state is unspecified. Consider the cost function

$$J = \int_0^{t_f} \frac{1}{2}\tan^2\theta + C_I \,\mathrm{d}t \tag{39}$$

where C_I is a tuning parameter known as the cost-index, analogous to the parameter in commercial aircraft (see chapter 3). Note that in many practical applications, quadrotors should not reach large pitch angles since they may lose altitude due to limitations on the total thrust that can be generated. To maintain a constant height as $\theta \to \pi/2$, the thrust must approach ∞. By using $\tan\theta$ in the cost function, we can discourage the use of large pitch angles since $|\tan\theta| \to |\infty|$ as $|\theta| \to |\pi/2|$. Using equation (6), and solving for T, the dynamics can be re-written as:

$$\dot{v} = g\tan\theta \tag{40}$$

Let us make the substitution $u = \tan\theta$. We can now formulate the optimal control problem as follows:

$$\begin{aligned} J^* = \min_{u,t_f} \int_0^{t_f} \frac{1}{2}u^2 + C_I \, \mathrm{d}t \\ s.t. \; \dot{x} = v \\ \dot{v} = gu \\ x(0) = x_0, \; v(0) = 0 \\ x(t_f) = 0, \; v(t_f) = 0 \end{aligned} \tag{41}$$

where we assume that $x_0 < 0$ without loss of generality. The Hamiltonian for this OCP is

$$H = \frac{1}{2}u^2 + C_I + \lambda_1 v + \lambda_2 g u \tag{42}$$

The necessary conditions is

$$\frac{\partial H}{\partial u} = u + \lambda_2 g = 0 \tag{43}$$

which yields

$$u^* = -g\lambda_2 \tag{44}$$

and the sufficient condition is satisfied:

$$\frac{\partial^2 H}{\partial u^2} = 1 > 0 \tag{45}$$

Since the final time t_f is free, and the running cost is not an explicit function of time we can use the property from equation (20). Substituting the optimal control law from (44) into (42) yields the Hamiltonian along the optimal trajectory:

$$H^* = -\frac{g^2}{2}\lambda_2^2 + C_I + \lambda_1 v = 0 \tag{46}$$

Solving equation (46) for λ_2 yields:

$$\lambda_2 = \pm\frac{1}{g}\sqrt{2(C_I + \lambda_1 v)} \tag{47}$$

where, for a real solution, $C_I + \lambda_1 v \geq 0$. From Pontryagin's Maximum Principle (19b), the co-state dynamics are as follows:

$$\begin{aligned} \dot{\lambda}_1(t) &= -\frac{\partial H}{\partial x} = 0 \\ \dot{\lambda}_2(t) &= -\frac{\partial H}{\partial v} = -\lambda_1(t) \end{aligned} \tag{48}$$

Integrating (48) from t to t_f yields that λ_1 is an unknown constant and

$$\lambda_2(t) = \lambda_2(t_f) + \lambda_1 \cdot (t_f - t) \tag{49}$$

Taking the second time derivative of v, from the dynamics in (41), yields:

$$\ddot{v} = g\dot{u} = g^2\lambda_1 \tag{50}$$

Integrating equation (50):

$$\begin{aligned} \dot{v}(t) &= g^2\lambda_1 t + A \\ v(t) &= \frac{g^2}{2}\lambda_1 t^2 + At + B \\ x(t) &= \frac{g^2}{6}\lambda_1 t^3 + \frac{1}{2}At^2 + Bt + C \end{aligned} \tag{51}$$

where $C = x_0$, $B = v_0$, and $A = -g^2\lambda_2(0)$, with final position and velocity equal to zero. Applying the initial and final conditions of the velocity v in equation (47) yields:

$$\begin{aligned} \lambda_2(0) &= \pm\frac{1}{g}\sqrt{2C_I} \\ \lambda_2(t_f) &= \pm\frac{1}{g}\sqrt{2C_I} \end{aligned} \tag{52}$$

To accelerate forwards, the pitch angle θ must be greater than 0 at $t = 0$ and θ must be less than zero at $t = t_f$ to deccelerate such that $v(t_f) = 0$. Therefore (52) now becomes:

$$\begin{aligned} \lambda_2(0) &= -\frac{1}{g}\sqrt{2C_I} \\ \lambda_2(t_f) &= +\frac{1}{g}\sqrt{2C_I} \end{aligned} \tag{53}$$

Let us consider equation (49) at $t = 0$ and substitute the expressions for λ_2 from (53)

$$-\frac{1}{g}\sqrt{2C_I} = \frac{1}{g}\sqrt{2C_I} + \lambda_1 \cdot t_f \tag{54}$$

and solving for the final time yields:

$$t_f = -\frac{2}{\lambda_1} \cdot \frac{\sqrt{2C_I}}{g} \tag{55}$$

Note that the final time must be positive and therefore $\lambda_1 < 0$. Applying the final condition for $x(t)$ as given by equation (51) yields

$$0 = \frac{g^2}{6}\lambda_1 t_f^3 + \frac{g}{2}\sqrt{2C_I}t_f^2 + v_0 t_f + x_0 \tag{56}$$

and then substituting the expression for the final time in (55) allows us to solve for λ_1 as

$$\lambda_1 = \frac{\sqrt{2C_I}}{2gx_0} \pm 2\sqrt{\frac{-\sqrt{2C_I^3}}{3gx_0}} \tag{57}$$

Selecting the positive sign in equation (57) ensures that the final time will be positive for all $C_I > 0$ and $x_0 < 0$. Substituting (57) into (47) yields

$$\lambda_2 = \pm\frac{1}{g}\sqrt{2C_I + v\left(\frac{\sqrt{2C_I}}{gx_0} + 4\sqrt{\frac{-\sqrt{2C_I^3}}{3gx_0}}\right)} \tag{58}$$

Which yields the optimal control input from (44):

$$u^* = \begin{cases} \sqrt{2C_I + 2v\left(\frac{\sqrt{2C_I}}{2gx_0} + \sqrt{\frac{-4\sqrt{2C_I^3}}{3gx_0}}\right)}, & x < \frac{x_0}{2} \\ -\sqrt{2C_I + 2v\left(\frac{\sqrt{2C_I}}{2gx_0} + \sqrt{\frac{-4\sqrt{2C_I^3}}{3gx_0}}\right)}, & x > \frac{x_0}{2} \\ 0, & x = \frac{x_0}{2} \end{cases} \tag{59}$$

The switching law for u^* can be found by examining the change in sign of λ_2 in equation (47). The switching time will occur at the maximum velocity which corresponds to the position of the quad being at the midpoint $x_0/2$, due to the symmetry of the problem. This symmetry can be seen in the trajectory of the time dependent results of the previous section (see figures 2.4, 2.5, and 2.6) when the initial and final velocity are selected to be 0. The maximum velocity, v_{max}, can be found as follows:

$$\begin{aligned} \lambda_2 &= \pm\frac{1}{g}\sqrt{2(C_I + \lambda_1 v_{max})} = 0 \\ v_{max} &= -\frac{C_I}{\lambda_1} \end{aligned} \tag{60}$$

Substituting (57) into (60) yields:

$$v_{max} = -\frac{6gx_0\sqrt{C_I}}{3\sqrt{2} + \sqrt{-48g\sqrt{2C_I}x_0}} \tag{61}$$

Note that equation (61) allows for the selection of C_I dependent on a restricted value of v_{max}. This will enable a range of C_I's to be determined such that the maximum velocity is not exceeded. A v_max can be selected according to limitations in the thrust and pitch angle that a quadrotor can generate or attain respectively. One may also consider the maximum allowable pitch angle of the quadrotor, and select the appropriate C_I according to equation (53) considering that $u = g\lambda_2$ = $\tan\theta$ (the largest pitch angles occur at $t = 0$ and $t = t_f$). As the value of C_I increases to very large values, the proposed problem will approach a minimum time problem and the commanded pitch angle will therefor approach 90 degrees.

λ_1	$\frac{\sqrt{2C_I}}{2gx_0} + 2\sqrt{\frac{-\sqrt{2C_I^3}}{3gx_0}}$
λ_2	$\pm\frac{1}{g}\sqrt{2C_I + v\left(\frac{\sqrt{2C_I}}{gx_0} + 4\sqrt{\frac{-\sqrt{2C_I^3}}{3gx_0}}\right)}$
t_f	$-\frac{2}{\lambda_1} \cdot \frac{\sqrt{2C_I}}{g}$
u^*	$\mp\sqrt{2C_I + 2v\left(\frac{\sqrt{2C_I}}{2gx_0} + \sqrt{\frac{-4\sqrt{2C_I^3}}{3gx_0}}\right)}$
v_{max}	$-\frac{6gx_0\sqrt{C_I}}{3\sqrt{2}+\sqrt{-48g\sqrt{2C_I}x_0}}$

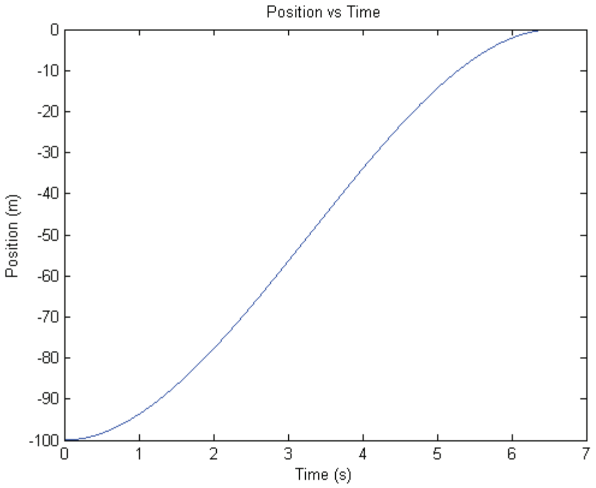


Figure 2.9: Position of the quadrotor as a function of time.

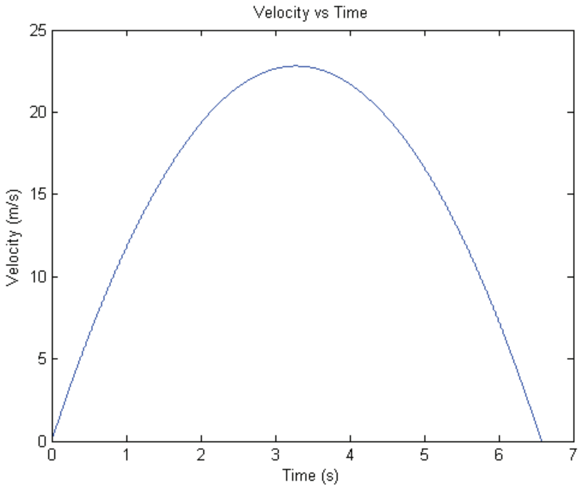


Figure 2.10: Velocity of the quadrotor as a function of time.

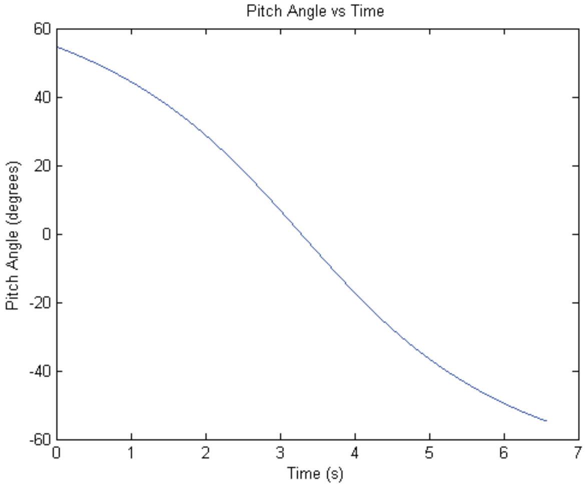


Figure 2.11: Pitch angle of the quadrotor as a function of time.

Figures 2.9, 2.10, and 2.11 illustrate the trajectory and pitch angle of a simulation of a quadrotor travelling 100 meters with a cost index (C_I) of 1. The quadrotor's position and velocity both have terminal values of zero and the quadrotor's velocity is symmetric about $tf/2$ as expected.

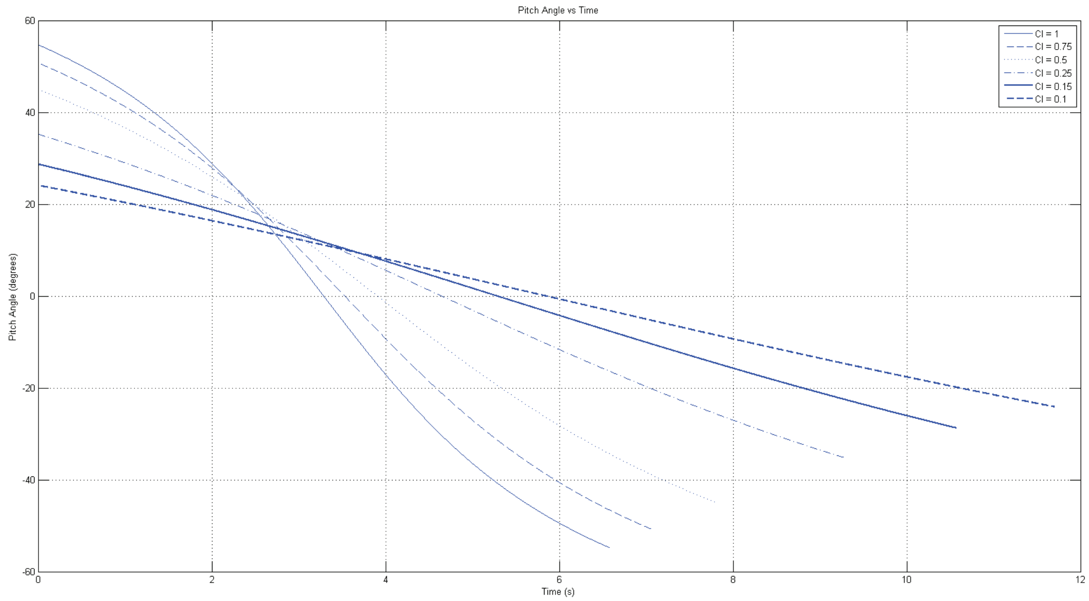


Figure 2.12: Pitch angle of the quadrotor for different values of C_I.

Figure 2.12 illustrates that smaller values of C_I lead to a smaller magnitude in the pitch angle and a longer final time t_f.

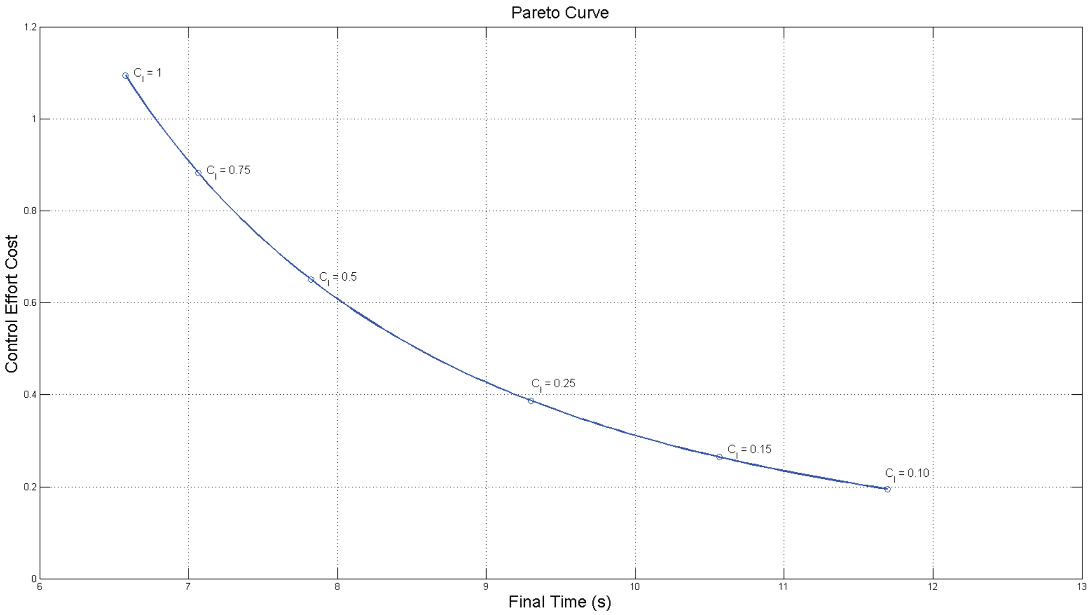


Figure 2.13: Pareto curve illustrating the trade-off between the accrued cost of control effort ($\int_0^{t_f} \frac{1}{2} \tan^2 \theta \, \mathrm{d}t$) and the final time ($t_f$) when varying C_I.

Figure 2.13 presents the trade-off between accrued control effort and final time. The plot gives insight into the benefit for choosing different values of C_I when considering both the accrued control effort and final time. Although terminal controllers (those which direct a dynamic system to a specified terminal state) have been present for many years, the terminal controllers presented in chapter 5 of [43] are restricted to quadratic cost functions and terminal penalties on the state, whereas the problem studied in this section has a penalty on the final time not the final state. Therefore we cannot directly use the results of the terminal controller section from [43] to solve the problem presented in this section.

2.3 Fixed-wing Path Following

Let us now consider a path for a fixed-wing UAV. We expect that a larger fixed-wing aircraft will be subject to similiar airspace restrictions as current commercial airliners. Although chapter 3 describes a flight management system in more detail for a fixed-wing UAV, let us consider a simplified case where a path between two points is given and the goal is to follow the given path.

Pilots in commercial aircraft will often use the LNAV (lateral navigation) and VNAV (vertical navigation) FMS capabilities to provide references to the auto-pilot. The lateral navigation component of an FMS provides guidance in the lateral plane according to the flight plan. To construct the LNAV module in the software FMS, we must first define a few navigational terms which are used in the lateral guidance of the aircraft. The cross-track error is the shortest distance between the aircraft and the active leg. The along-track distance is the distance along the track of the aircraft if the cross-track error was zero (see Figure 2.3). There are many terms that describe different angles in the lateral plane, these include: course, track, bearing and heading. The desired track, or course, is the angle measured clockwise from true north to the path and is independent of the aircraft. Bearing is measured clockwise from true north to the vector which starts at the aircraft and ends at the desired waypoint. Heading is the clockwise angle formed between true north and the nose of the aircraft. Figure 2.3 illustrates these angles. Note that pilots will generally use the angles measured from magnetic north, not true north, therefore the magnetic declination (also known as magnetic variation) must be used to convert between true and magnetic north:

$$M_N = T_N + D \tag{62}$$

where M_N is magnetic north, T_N is true north, and D is the declination in degrees West.

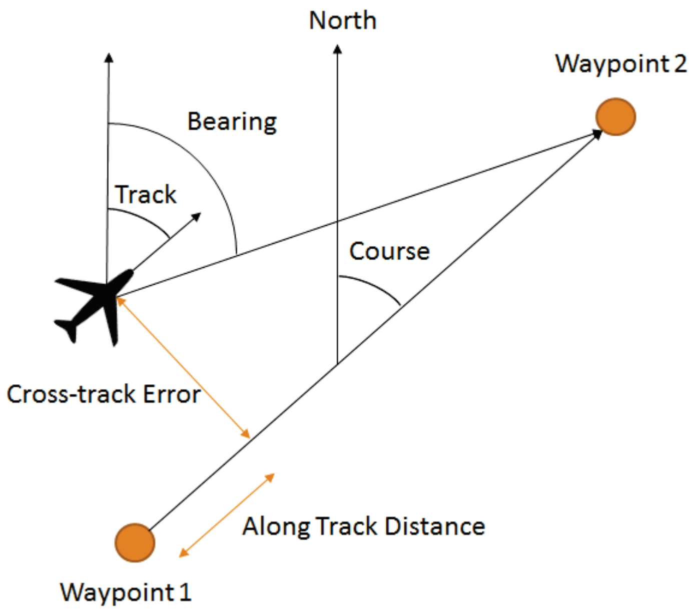


Figure 2.14: Illustration of lateral guidance variables.

To guide the aircraft in the lateral plane along the active leg, we must develop a controller which provides a desired roll angle to the auto-pilot. Reference [44] provides a general form for the desired roll angle as follows:

$$\phi_{des} = k_{xtrk}e_{xtrk} + k_{trk}e_{trk} + \phi_o \tag{63}$$

where k_{xtrk} is the cross-track error gain, e_{xtrk} is the cross-track error, k_{trk} is the track error gain, e_{trk} is the track error, and ϕ_o is the nominal roll angle (i.e. turn angle). We will consider an alternative control law for the desired roll angle under the assumption that we are interested in tracking the path locally. With this assumption, the equations of motion in the horizontal plane over a flat Earth are given by [45]

$$\begin{aligned}
\dot{x} &= v\cos\psi \\
\dot{y} &= v\sin\psi \\
\dot{v} &= \frac{g}{W}(T\cos\alpha - D) \\
\dot{\psi} &= \frac{g}{Wv}(T\sin\alpha + L)\sin\mu \\
\dot{\gamma} &= \frac{g}{Wv}(T\sin\alpha\cos\mu + L\cos\mu - W) = 0 \\
\dot{W} &= -S_{FC}T
\end{aligned} \tag{64}$$

where ψ is the heading angle, μ is the bank angle, assuming that there is no sideslip. Let us also assume that the angle of attack α is small, the bank angle μ is small, the term $T\sin\alpha$ is much smaller than the lift L and weight W, and the aircraft is flying at a constant speed ($\dot{v} = 0$). These assumptions are considered reasonable for commercial aircraft and lead to a simplification of equation (64) as

$$\begin{aligned}
\dot{x} &= v\cos\psi \\
\dot{y} &= v\sin\psi \\
T &= D \\
\dot{\psi} &= \frac{g}{Wv}L\sin\mu \\
L &= W \\
\dot{W} &= -S_{FC}T
\end{aligned} \tag{65}$$

The centripetal force during a turn is caused by the component of the lift in the horizontal plane, as illustrated in Fig. 2.3, providing the following constraint

$$L\sin\mu = \frac{W}{g}\frac{v^2}{r} \tag{66}$$

where r is the radius of the turn.

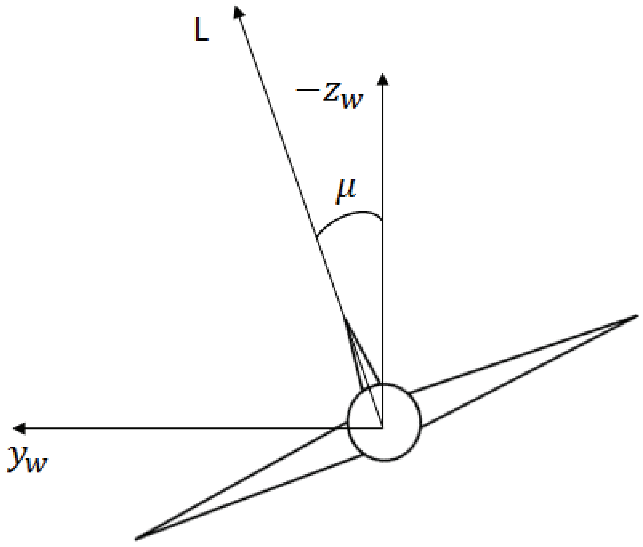


Figure 2.15: Horizontal component of the lift providing the centripetal acceleration during a turn.

Without loss of generality, we can represent the lateral guidance of the aircraft to the desired path as the convergence of the aircraft to the x_c axis, where the x_c axis aligns with the desired path (see Fig. 2.3). We desire that $y_{ac} = 0$ and that the aircraft's heading tracks the desired heading ($\psi = \psi_d$). We now present the control law stated as a theorem following closely the work of [46].

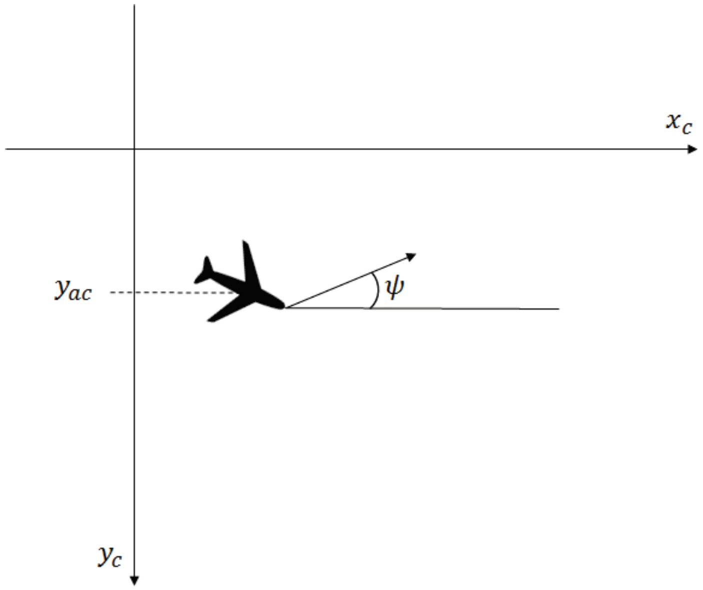


Figure 2.16: The desired path is along the x_c axis, with ψ being the relative heading between the aircraft and the x_c axis. The value of y_{ac} represents the cross-track error.

Theorem 1. *Consider a desired heading $\psi_d = -g_\delta(y_{ac})$ where y_{ac} is the y_c coordinate of the aircraft, g_δ is class C^1, the aircraft dynamics in equation (65) and the following bank angle command*

$$\sin\mu = \frac{Wv}{gL}\left(\frac{\partial\psi_d}{\partial y_{ac}}v\sin\psi - \lambda_\psi y_{ac}v\frac{\sin\psi - \sin\psi_d}{\psi - \psi_d} - k\lambda_\psi v(\psi - \psi_d)\right) \tag{67}$$

where $\lambda_\psi > 0$ and $k > 0$ are constant parameters, and with the constraints $g_\delta(0) = 0$ and $y_{ac}\sin(g_\delta(y_{ac})) > 0$, $\forall y_{ac} \neq 0$, $y_{ac}\sin(g_\delta(y_{ac})) = 0$ for y_{ac} = 0. Then y_{ac} and ψ converge to zero asymptotically.

Proof. Let us consider the following candidate Lyapunov function from [46]

$$V = \frac{1}{2}y_{ac}^2 + \frac{1}{2\lambda_\psi}\left(\psi - \psi_d\right)^2 \tag{68}$$

where y_{ac} is the y-coordinate of the aircraft in the course axes, $\lambda_\psi > 0$, ψ is the relative heading of the aircraft to the x_c axis and ψ_d is the relative desired heading as a function of y_{ac}. Both ψ and ψ_d are measured from the x_c axis in the clockwise direction. For V to be a Lyapunov function, with state vector $\mathbf{x} = \begin{bmatrix} y_{ac} & \psi \end{bmatrix}^T$, the following conditions must be satisfied [47]:

$$\begin{aligned} &1.\, V(\mathbf{x}) > 0,\ \forall \mathbf{x} \neq 0 \\ &2.\, V(\mathbf{x}) = 0,\ \mathbf{x} = 0 \\ &3.\, \dot{V}(\mathbf{x}) < 0,\ \forall \mathbf{x} \neq 0 \\ &4.\, \dot{V}(\mathbf{x}) = 0,\ \mathbf{x} = 0 \end{aligned} \tag{69}$$

The function V in (68) is positive, and zero when $y_{ac} = 0$ and $\psi = \psi_d$. If we enforce that $\psi_d(0) = 0$ (as stated in the theorem: $\psi_d = -g_\delta(y_{ac})$ and $g_\delta(0) = 0$) then V meets the requirements of the first two conditions in (69). The time derivative of equation (68), yields

$$\dot{V} = y_{ac}\dot{y}_{ac} + \frac{1}{\lambda_\psi}(\psi - \psi_d)(\dot{\psi} - \dot{\psi}_d) \tag{70}$$

Substituting the dynamics from (65) into (70)

$$\dot{V} = y_{ac} v \sin\psi + \frac{1}{\lambda_\psi}(\psi - \psi_d)(\frac{g}{Wv} L \sin\mu - \dot{\psi}_d) \tag{71}$$

We must now select a control input, $\sin\mu$, to ensure $\dot{V}$ is negative definite as required in (69). Using the control input proposed in the theorem and substituting it into equation (71) yields

$$\dot{V} = y_{ac} v \sin\psi_d - kv(\psi - \psi_d)^2 \tag{72}$$

which requires that the speed of the aircraft $v > 0$, $y_{ac}\sin\psi_d > 0,\ \forall y_{ac} \neq 0$ and $y_{ac}\sin\psi_d = 0,\ y_{ac} \,|\, y_{ac} = 0$, and $k > 0$ for V to be a Lyapunov function. □

Let us consider the sigmoid function presented in [46] which satisfies the conditions of the theorem:

$$g_\delta(y_{ac}) = \psi_a \frac{e^{2k_\delta y_{ac}} - 1}{e^{2k_\delta y_{ac}} + 1},\ 0 \leq \psi_a < \pi \tag{73}$$

where $k_\delta > 0$ and the profile of the desired heading with respect to y_{ac} can be tuned using k_δ. Figure 2.3 illustrates the profile of equation (73) for a few values of k_δ, where $y_a c$ is the cross-track error and the relative heading is the desired heading ψ_d. As y_{ac} increases in magnitude, $|g_\delta| \to \psi_a$. With $\psi_d = -g_\delta$, we can select $\psi_a = \pi/2$ to provide a desired heading that is perpendicular to the course when the aircraft is relatively ”far” from the desired track.

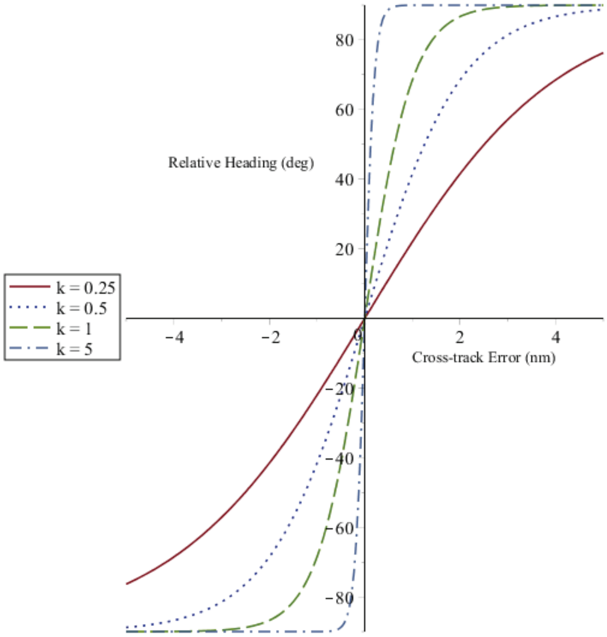


Figure 2.17: Profile of the sigmoid function for various gains k_δ.

The work presented in [46] provides trajectory tracking for wheeled robots, whereas this section has used the technique from [46] to apply similar trajectory tracking for fixed-wing UAVs and aircraft.

Chapter 3

Software Flight Management System

Most large commercial aircraft have a flight management system (FMS) which performs many critical functions, such as [44]:

- Estimating states of the aircraft (i.e. position, speed, etc.)
- Constructing a flight plan
- Trajectory prediction
- Lateral and vertical guidance
- Performance functions (i.e. optimal cruise speed)

Commercial flight management systems are required to adhere to strict requirements (such as those presented in [48]). We aim at replicating some functionality, as listed in the requirements below, which may be extended in future work. For the prototype developed in this thesis, we do not impose any restrictions or regulations upon the software FMS (SFMS). For example, we do not consider the software certification and we do not consider airspace restrictions for the lateral navigation and guidance.

3.1 Software Flight Management System Overview

The SFMS must provide the following functionality:

- Interface with the Marinvent Piaggio Avanti flight simulator housed at Concordia University using ARINC 429 and 834 protocols
- Compute and display the optimal target airspeed for cruise
- Perform lateral navigation and guidance to follow a given flight plan
- The user must be able to create a flight plan from a navigation database

To address these requirements the software will be programmed in a modular architecture, with a clear mapping of a real FMS to the software FMS. Fig. 3.1 indicates the block diagram components of an FMS that will be addressed in the software FMS.

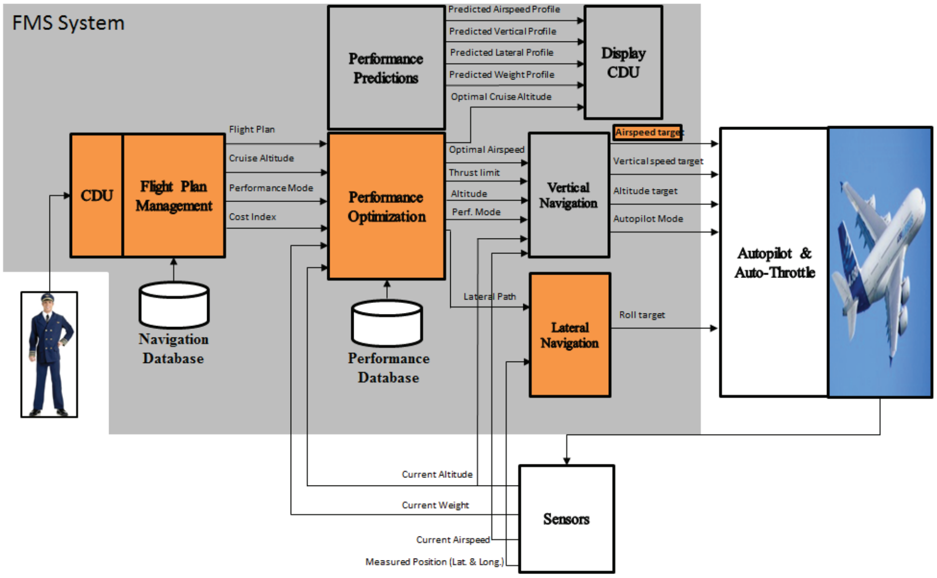


Figure 3.1: FMS block diagram with highlighted blocks relating to the software FMS [1].

Let us first examine how each component will be translated to a software module. Using the diagram in Fig. 3.1 as reference, the central display unit (CDU) will be the user interface (UI) on a tablet or laptop. The flight plan management will construct a flight plan from the navigational waypoints entered by the user. The cost index entered by the user will be utilized in the ”performance” module to determine the airspeed during cruise (which is highlighted as the airspeed target). The lateral navigation will handle the lateral path determined by the waypoints in the flight plan and output a desired roll command to the pilot or autopilot. The software FMS is divided into modules as illustrated in Fig. 3.2. The navigation module is divided into two sub-modules, one to parse the navigation database and the other to perform lateral navigation and guidance, labelled as ”Parser” and ”Lateral” respectively.

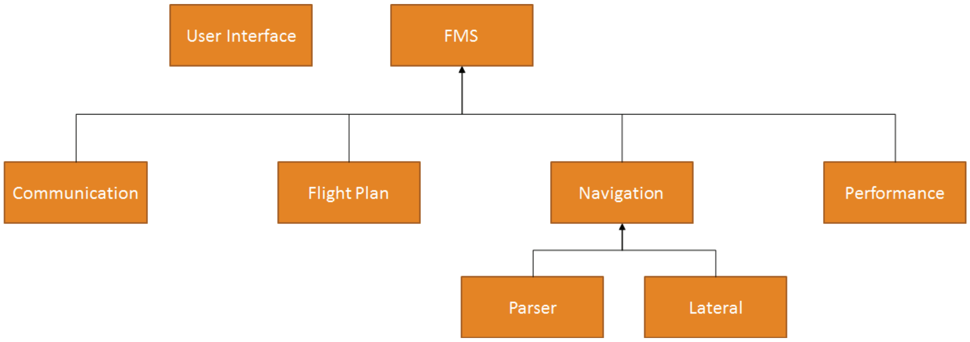


Figure 3.2: Sofware FMS module block diagram.

The communication module will perform the encoding/decoding according to the ARINC 429 and 834 protocols, as well as setup the network connection with the AID. The flow chart representing the program flow is illustrated in Fig. 3.3. Upon launching the SFMS, the waypoints from the navigation database are loaded into memory along with the aircraft parameters. A communication thread creates the network connection and requests/sends data using the ARINC 429/834 protocol. In parallel with the communication, the user can enter a flight plan. Note that the lateral guidance does not engage until the user has loaded the flight plan. A roll angle of 0 degrees is sent as a reference until the lateral guidance is activated. The user will also be able to specifiy a cost index and the optimal target indicated airspeed will be displayed as determined by equation (93).

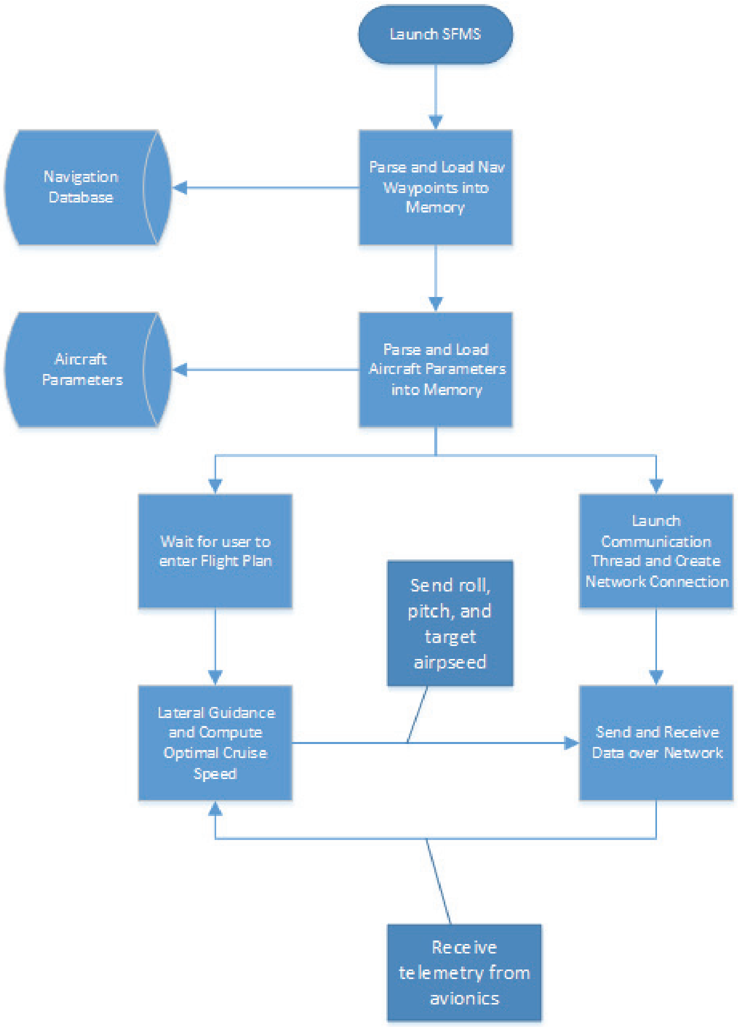

Launch SFMS
Navigation Database
Parse and Load Nav Waypoints into Memory
Aircraft Parameters
Parse and Load Aircraft Parameters into Memory
Wait for user to enter Flight Plan
Launch Communication Thread and Create Network Connection
Send roll, pitch, and target airpseed
Lateral Guidance and Compute Optimal Cruise Speed
Send and Receive Data over Network
Receive telemetry from avionics

Figure 3.3: The software FMS high level program flow chart.

The rest of this chapter describes the prototype software flight management system that was developed as a research platform in further detail.

3.2 Preliminaries

3.2.1 Longitudinal Equations of Motion of an Aircraft

The material in this section is based on [45], [49], and [2]. An aircraft is subject to aerodynamic forces during flight due to the flow of air around the body of the aircraft. To derive the equations of motion in the longitudinal plane, a few assumptions must be made:

- The Earth is flat and non-rotating.
- The atmosphere is at rest relative to the Earth.
- The aircraft has a right-left plane of symmetry and is modeled as a variable-mass particle.
- The forces acting on the aircraft act at its center of gravity.

We can now define four coordinate systems using these assumptions. The ground axes are fixed on the ground plane at mean sea level with the xz plane as the vertical plane. The horizon coordinate system is positioned at the aircraft center of gravity and is parallel to the ground axes (i.e. it does not rotate with the aircraft body). The wind coordinate system is also located at the center of gravity of the aircraft, with the x_w axis pointing in the direction of the velocity of the aircraft. The body axes are fixed to the center of gravity of the aircraft and will rotate with the body.

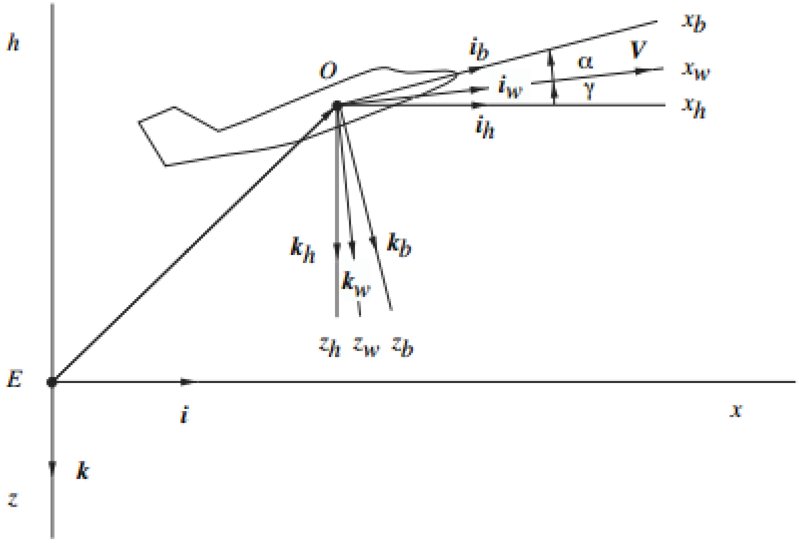


Figure 3.4: Coordinate systems in longitudinal plane [2].

The flight path angle γ is the angle between the wind axes and the horizon axes as illustrated in Fig. 3.2.1. The angle between the body axes and the wind axes is the angle of attack α. The longitudinal equations of motion of an aircraft, with speed v and weight W, in the ground coordinate frame are

$$\begin{aligned}
\dot{x} &= v\cos\gamma \\
\dot{h} &= v\sin\gamma \\
\dot{v} &= \frac{g}{W}(T\cos\alpha - D - W\sin\gamma) \\
\dot{\gamma} &= \frac{g}{Wv}(T\sin\alpha + L - W\cos\gamma) \\
\dot{W} &= -S_{FC}T
\end{aligned} \tag{74}$$

Note that we are assuming that the thrust vector generated from the propulsion is aligned with the body axis x_b. The forces due to the lift and drag are expressed as follows:

$$\begin{aligned}
L &= \frac{1}{2}C_L(\alpha, R_e, M)\rho S v^2 \\
D &= \frac{1}{2}C_D(\alpha, R_e, M)\rho S v^2
\end{aligned} \tag{75}$$

where the coefficients C_L and C_D are functions of the angle of attack α, Reynolds number R_e, and the Mach number M. The lift and drag coefficients can be further expanded as

$$\begin{aligned}
C_L &= C_{L_0} + C_{L_\alpha}\alpha \\
C_D &= C_{D_0} + C_{D_i}C_L^2
\end{aligned} \tag{76}$$

where C_{L_0} is the incompressible lift coefficient, C_{L_α} is the angle of attack lift coefficient , C_{D_0} is the profile drag coefficient, and C_{D_i} is the induced drag coefficient due to lift. Let us now consider some assumptions that can be made for an aircraft in quasi-steady flight during cruise:

- $\gamma = 0$
- $\dot{\gamma} = 0$
- $\dot{v} = 0$
- $\dot{h} = 0$
- α is small
- $|T\sin\alpha| << W$

Applying these assumptions to equation (74) results in the following simplified longitudinal equations of motion:

$$\begin{aligned}
\dot{x} &= v \\
T &= D \\
L &= W \\
\dot{W} &= -S_{FC}T
\end{aligned} \tag{77}$$

3.2.2 Aircraft Navigation

This section is based on the material in [50][51][52][53]. The primary reference frames used in aircraft navigation are Earth Centered Inertial (ECI), Earth Centered Earth Fixed (ECEF), the Geodetic reference frame, and the local tangent plane (see Fig. 3.2.2). The ECI reference frame is considered fixed, with the origin at the center of the Earth aligned with the equatorial plane, relative to the distant stars with $\mathbf{x}_i$ aligning with a celestial reference point, $\mathbf{z}_i$ aligned with the axis of rotation in the direction of North, and $\mathbf{y}_i$ chosen such that $\mathbf{x}_i \times \mathbf{y}_i = \mathbf{z}_i$. ECEF is aligned with the ECI reference frame along the z_i axis but rotates with the Earth, where the $\mathbf{x}_e$ axis connects the center of the Earth with the intersection between the Greenwich meridian and the equator. The Geodetic reference frame also rotates with the Earth and has its origin at the center of the equatorial plane. The Geodetic coordinates are defined by two angles, latitude ϕ and longitude λ, and a height above the Earth h. Latitude is measured from the equator and is considered positive in the northern hemisphere and negative in the southern hemisphere. Longitude is measured from the Greenwich Meridian and is positive in the eastern direction and negative in the western direction. Latitude is constrained between -90 and 90 degrees, while longitude is in the range of -180 and 180 degrees. The heading of an aircraft is considered relative to true north and is increasing in the clockwise direction.

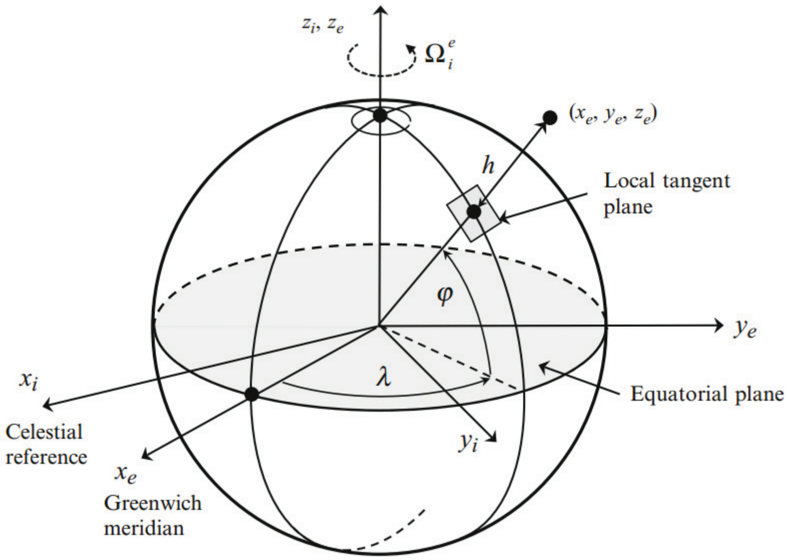


Figure 3.5: ECEF, ECI, Geodetic, and local tangent plane reference frames [3].

Although we considered a flat Earth when deriving the longitudinal dynamics of an aircraft locally, we must consider a spherical Earth for the purpose of generating the path between points on the Earth. Using the approximation of the Earth as a sphere allows us to simplify the calculations required for navigation and provides sufficient accuracy for the prototype software FMS. We will consider flight paths which do not come close to the poles of the Earth.

The distance between two points and the departure course are of particular interest to aircraft navigation. The shortest path between two points on a sphere is a geodesic, which is a path that lies on a great circle. A great circle is defined as a circle with the same radius as the Earth and partitions the sphere into two equal parts. To derive the equations for the distance and course, one creates a spherical triangle on the surface of the Earth (see Fig. 3.6).

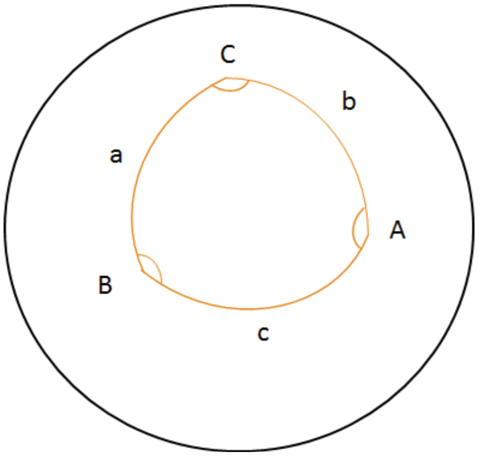


Figure 3.6: A spherical triangle created by intersecting geodesics.

Using spherical geometry [53], the relationship between the length of the sides and angles of a spherical triangle is given by Napier's rules as follows:

$$\begin{aligned}\cos a &= \cos b \cos c + \sin b \sin c \cos A \\ \cos b &= \cos a \cos c + \sin a \sin c \cos B \\ \cos c &= \cos a \cos b + \sin a \sin b \cos C\end{aligned} \tag{78}$$

Note that the length of the sides of a spherical triangle are in radians since the equations are derived on a unit sphere and therefore the distance of the arc length will be equal to angle created by the arc. To extend the results to navigation on a spherical model of the Earth, the arc lengths can be represented in distance units by multiplying the arc length by the radius of the Earth. Equation (78) can be further manipulated, under the assumption that the six angles a, b, c, A, B, and C lie in the range $[0, \pi]$, to yield the law of sines:

$$\frac{\sin a}{\sin A} = \frac{\sin b}{\sin B} = \frac{\sin c}{\sin C} \tag{79}$$

Let us consider the two points A and B, with geodetic coordinates (λ_1, ϕ_1) and (λ_2, ϕ_2), using the spherical model of the Earth. We will consider both points to have the same height (i.e. on the surface of the Earth). Assigning point C in Fig. 3.6 to be the North Pole and points A and B as the two points of interest, we can make the following substitutions

$$C = \phi_2 - \phi_1 \tag{80}$$

and

$$\begin{aligned} a &= \frac{\pi}{2} - \lambda_2 \\ b &= \frac{\pi}{2} - \lambda_1 \end{aligned} \tag{81}$$

The arc length c is the distance between the two points of interest, which will be denoted as d (see Fig. 3.7).

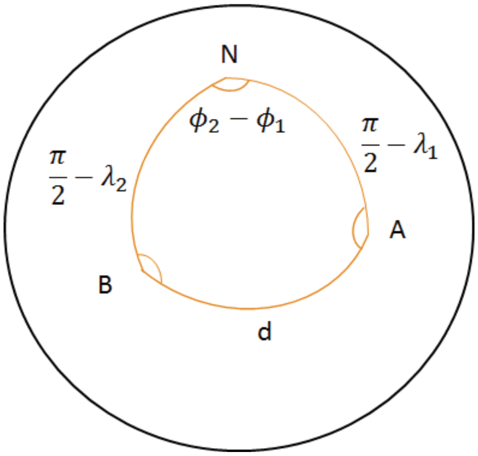


Figure 3.7: A spherical triangle created by intersecting geodesics and applying the geodetic coordinates.

Applying equation (78) to the spherical triangle illustrated in Fig. 3.7 and solving for arc length d yields

$$d = \cos^{-1}\left(\cos(\frac{\pi}{2} - \lambda_2)\cos(\frac{\pi}{2} - \lambda_1) + \sin(\frac{\pi}{2} - \lambda_2)\sin(\frac{\pi}{2} - \lambda_1)\cos(\phi_2 - \phi_1)\right) \tag{82}$$

To find the true course at which to depart from (λ_1, ϕ_1) to (λ_2, ϕ_2), we apply equation (79) and solve for the angle A:

$$A = \sin^{-1}\left(\frac{\sin(\frac{\pi}{2} - \lambda_2)\sin(\phi_2 - \phi_1)}{\sin d}\right) \tag{83}$$

Note that (83) is not well defined for $d = n\pi$ where n is any integer, and that the true course will vary along the geodesic. The literature addresses implementation details such as the prevention of inaccuracies near the poles when performing certain computations. However, these will not be discussed in the work presented here.

3.2.3 Optimal Cruise Speed

We cannot determine the optimal cruise speed without first defining the functional to be optimized. From the perspective of a commercial airline, it would likely be in terms of the cost of the flight. The cost of a flight can be divided into time-related costs (wages, maintenance, etc.) and the cost of burned fuel. The cost index (C_I) represents the ratio of the time-related costs and the cost of fuel. Therefore a large C_I will put more emphasis on the time-related costs and a small C_I will put more emphasis on the cost of fuel. A value of $C_I = 0$ considers only the cost of fuel with no consideration for the time of flight and as $C_I \to \infty$ the opposite is true. The work in [28] has found an analytic expression for the cruise speed for a given aircraft and a given cost index, by formulating the problem as an optimal control problem. The performance measure to be minimized in [28] is

$$J = \int_{t_c}^{t_d} f_f + C_I \, \mathrm{d}t \tag{84}$$

where t_c is the time when the aircraft begins to cruise (top of climb), t_d is the unspecified time when the aircraft ends cruise (top of descent), and f_f is the fuel flow rate. Let us consider the simplified longitunidal equations of motion from (77). Using the condition in steady flight that $L = W$ and equation (76), the lift coefficient C_L can be expressed as follows:

$$C_L = \frac{2W}{\rho S v^2} \tag{85}$$

and is substituted into the expression for the drag coefficient in equation (76) to yield

$$C_D = C_{D_0} + C_{D_i} \left(\frac{2W}{\rho S v^2} \right)^2 \tag{86}$$

Equation (86) can then be substituted into (75) to yield the following force due to drag

$$D = \frac{1}{2} C_{D_0} \rho S v^2 + 2 C_{D_i} \frac{W^2}{\rho S v^2} \tag{87}$$

assuming that the aircraft is flying below the drag divergence Mach number. With the constraint $T = D$ we can reduce the longitudinal dynamics in (77) as follows

$$\begin{aligned} \dot{x} &= v \\ \dot{W} &= -S_{FC} D \end{aligned} \tag{88}$$

The optimal FMS control problem can now be expressed as [28]

$$\begin{aligned} J^* = &\min_{v,\, t_d} \int_{t_c}^{t_d} S_{FC} D + C_I \, \mathrm{d}t \\ & s.t. \\ & \quad \dot{x} = v \\ & \quad \dot{W} = -S_{FC} D \\ & \quad x(t_c) = x_c,\, x(t_d) = x_d \\ & \quad W(t_c) = W_c \end{aligned} \tag{89}$$

where W_c is the weight of the aircraft at the top of climb, x_c is the position of the aircraft at the top of climb, and x_d is the position of the aircraft at the top of descent. The solution to the optimal control problem in (89) is found to be [28]

$$v = \sqrt{\frac{C_I + \sqrt{C_I^2 + 12(1 - J_W^*)^2 S_{FC}^2 C_{d0} C_{di} W^2}}{(1 - J_W^*) S_{FC} C_{d0} \rho S}} \tag{90}$$

where the dynamics of J_W^* are

$$\dot{J}_W^* = (J_W^* - 1) \frac{4 S_{FC} C_{di} W}{\rho S v^2} \tag{91}$$

with the following known final value

$$J_W^*(t_d) = 0 \tag{92}$$

In [28], it is shown that $0 \leq J_W^* < 1$ and can be approximated as zero for all time. This approximation becomes less accurate for longer flights. With the substitution of $J_W^* = 0$ into (90), the sub-optimal solution is

$$v \approx \sqrt{\frac{C_I + \sqrt{C_I^2 + 12S_{FC}^2 C_{d0} C_{di} W^2}}{S_{FC} C_{d0} \rho S}} \tag{93}$$

The sub-optimal speed in (93) will be used in the software FMS as the reference speed for the pilot.

3.3 The ARINC 429 Communication Protocol

The ARINC 429 Specification [54][4] describes how avionics systems should interact with one another, i.e. the electrical characteristics and the bit structure. The avionics devices (such as a GPS) are designated as line replaceable units, or LRUs. The LRUs are connected using a shielded twisted pair wire and only support uni-directional communication. Each transmitter supports up to 20 receivers on a single twisted pair wire. For bi-directional communication, such as serial receive and transmit, a second twisted pair wire is required. Fig. 3.8 presents the different topologies found in ARINC 429, with the ”Star” and ”Bus-Drop” topologies being the most commonly used.

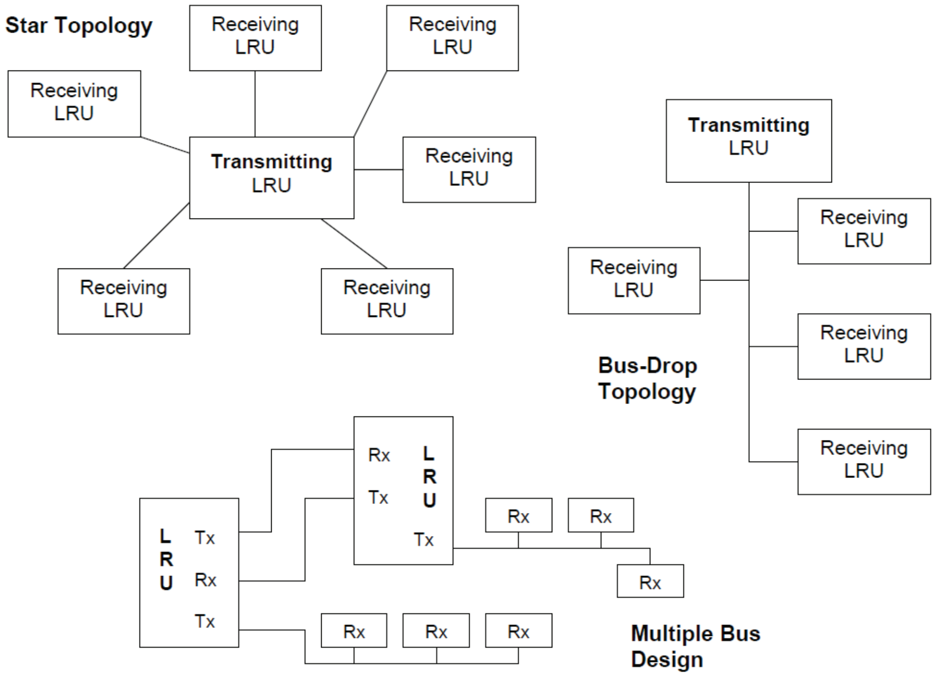
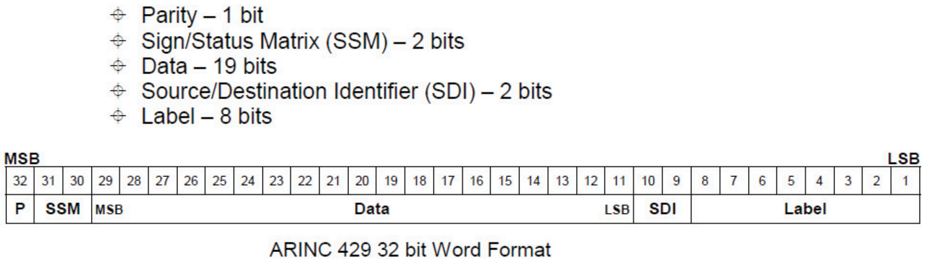


Figure 3.8: ARINC 429 connection topologies [4].

A message in ARINC 429 commonly consists of a single 32 bit word, although multiple 32 bit words are possible in a single message (see Fig. 3.9). The label and the parity bit are the only necessary overhead bits for each 32 bit word, while the sign/status matrix (SSM) and source/desitnation identifier (SDI) bits can be used to provide extra precision. The use of the SSM and SDI bits are specific to the label being used and the documentation (see [54]) should be referenced for the use of the labels. The data word can be coded in either binary, binary coded decimal, or alphanumeric data according to ISO (International Organization for Standardization) alphabet No. 5 [54]. The label is decoded into octal format to match the standard.



- Parity – 1 bit
- Sign/Status Matrix (SSM) – 2 bits
- Data – 19 bits
- Source/Destination Identifier (SDI) – 2 bits
- Label – 8 bits

MSB LSB

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
P	SSM		MSB									Data									LSB	SDI		Label							

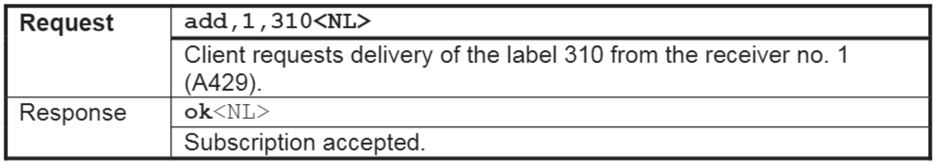
ARINC 429 32 bit Word Format

Figure 3.9: ARINC 429 word format [4].

3.4 The ARINC 834 STAP Communication Protocol

The ARINC 834 Specification [5] describes how to interface with an aircraft network using commercially available technology (i.e. laptop, tablet, etc). The ARINC 834 Specification provides three aircraft interface services: Generic Aircraft Parameter Service (GAPS), Simple Text Avionics Protocol (STAP) and Avionics Data Broadcast Protocol (ADBP). Currently only STAP provides both read and write access to avionics devices, and is therefore the only viable choice for the software FMS.

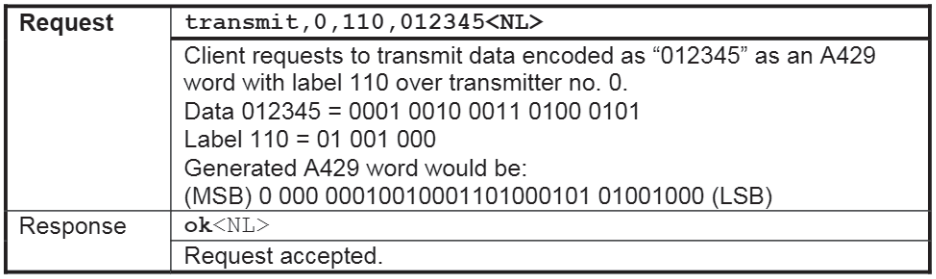
Marinvent Corporation has supplied a STAP program, Stapify, which the software FMS can connect to. Using Stapify, we are able to request data from the avionics (longitude, latitude, altitude, etc.), as well as send data (roll and pitch commands, and a target speed). The process of requesting data from the avionics functions as a subscription service. By subscribing to a particular label, the software FMS will receive the desired data periodically. Fig. 3.10 describes the syntax to request a subscription where the receiver number is the ID of the avionics component we would like to request the specified label from.



Request	**`add,1,310<NL>`**
	Client requests delivery of the label 310 from the receiver no. 1 (A429).
Response	`ok<NL>`
	Subscription accepted.

Figure 3.10: ARINC 834 syntax to add a subscription [5].

To transmit data one must first encode the data in ARINC 429 according to Fig. 3.11 and then wrap that data in ARINC 834 according to Fig. 3.11. This process is illustrated in Fig. 3.12.



Request	`transmit,0,110,012345<NL>`
	Client requests to transmit data encoded as "012345" as an A429 word with label 110 over transmitter no. 0. Data 012345 = 0001 0010 0011 0100 0101 Label 110 = 01 001 000 Generated A429 word would be: (MSB) 0 000 00010010001101000101 01001000 (LSB)
Response	`ok<NL>`
	Request accepted.

Figure 3.11: ARINC 834 syntax to transmit data [5].

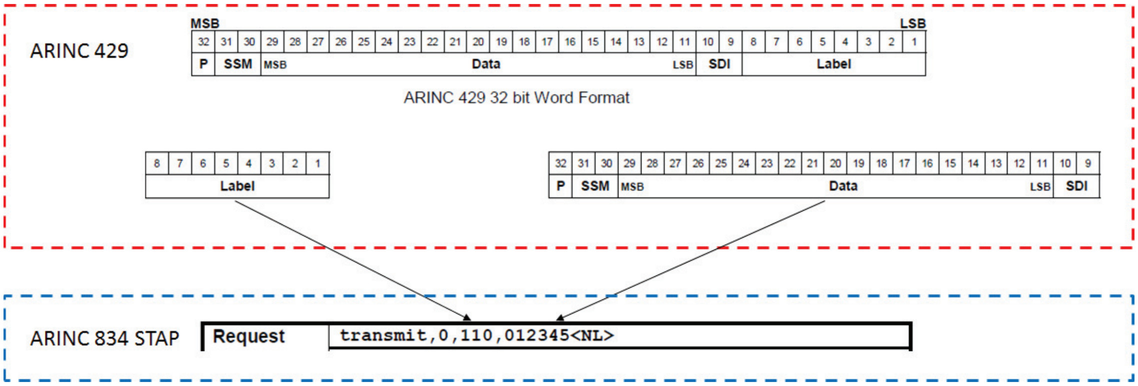
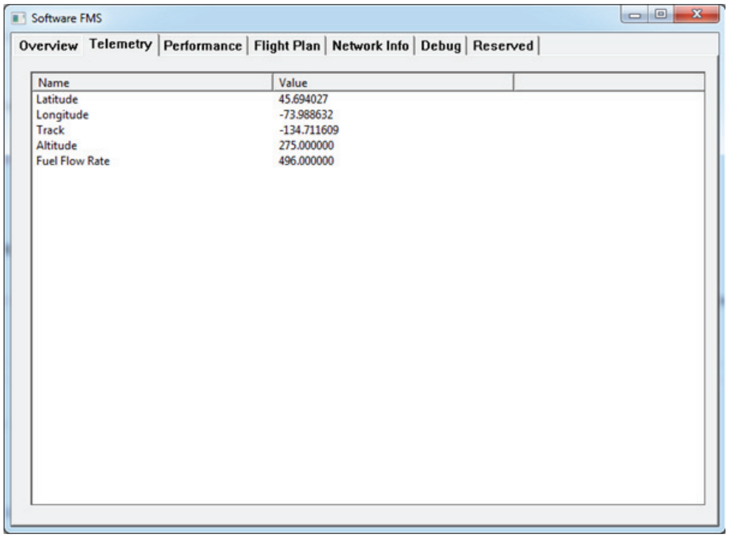


Figure 3.12: ARINC 429 wrapped in ARINC 834 [4][5].

Using the protocol described above, the software FMS is able to receive latitude, longitude, track, altitude and the fuel flow rate, as well as send a roll command, pitch command, and a target airspeed, in real time (as illustrated in Fig. 3.13).



Name	Value	
Latitude	45.694027	
Longitude	-73.988632	
Track	-134.711609	
Altitude	275.000000	
Fuel Flow Rate	496.000000	

Figure 3.13: Aircraft states are displayed by the software FMS in real time using the ARINC 429/834 Specifications.

3.5 Constructing a Flight Plan

A flight plan can be defined as the route an aircraft will take from a departure airport to a destination airport [44]. It may contain a variety of different waypoints and procedures, such as:

- SID (standard instrument departure) and STAR (standard arrival) procedures
- Airways
- Pre-defined routes
- Fixes (en route waypoints, navaids, nondirectional beacons, terminal waypoints, airport reference points, runway thresholds)
- Crew-defined fixes (i.e. lat/lon waypoints)

SID and STAR are procedures for taking-off and landing, respectively. These procedures are stored in a database that can be accessed by the FMS. While the aircraft is stationary at the airport gate before take-off, the pilot will usually enter the flight plan into the FMS. The pilot will be required to provide the departure procedure and enter the identifiers for the waypoints along the flight. The software FMS will be restricted to a database of navigational waypoints taken from X-Plane ignoring SIDs, STARs, airways, and restricted airspace, to reduce the complexity of the flight plan software module. The user of the SFMS can enter the identifier for an airport or navigational waypoint, and the software FMS will then compute great circles connecting navigational waypoints from the database to provide the departure course from each waypoint and distance between them.

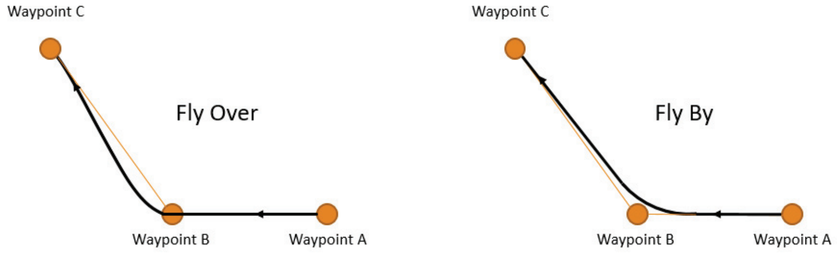


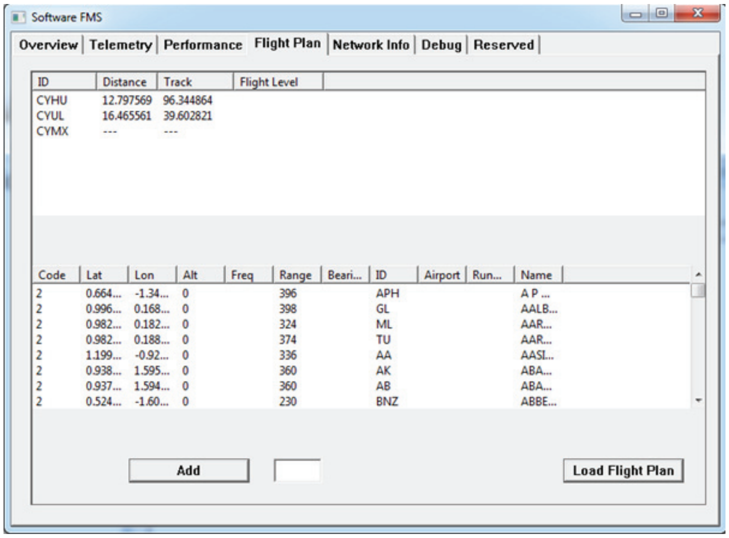
Figure 3.14: Fly over and fly by turn maneuvers for an aircraft traveling from waypoint A to C, passing through waypoint B. The black line is the aircraft path and the orange line is the direct path between waypoints.

The path between two waypoints will be defined as a ”leg”. A flight plan is generally made up of many legs and the FMS must compute the transitions between adjoining legs. As stated by the FAA [48]: ”If the course outbound from a fix differs from the course inbound to the fix (courses measured at the fix), a turn is indicated”. Constructing a turn requires the consideration of a pilot’s reaction time as well as the delay in the roll response of the aircraft. Although the construction of a turn is a complex procedure, we will consider a simplified case for the software FMS. Two common turn types will be considered, the ”fly over” and ”fly by” turn maneuvers. The fly over turn maneuver only initiates a turn once the aircraft has flown over the target waypoint. This causes an overshoot of the next lateral path (see Fig. 3.14). The fly by turn maneuver anticipates the next turn and will by-pass the waypoint to perform a turn which prevents any overshoot of the next leg (see Fig. 3.14). The fly by turn maneuver is the prefered method when possible since it provides the shortest path when transistioning between legs. Note that the turn maneuver is dependent on many variables, such as airspace restrictions and the magnitude of the change in heading required.

Reference [48] describes in detail many of the different situations that can be encountered.

3.6 Experimental Results Using a Flight Simulator

Using Marinvent Corporation's flight simulator, we were able to validate the prototype software FMS. Let us consider a flight in the Montreal area of Quebec, Canada. The test flight will consist of three waypoints to test the fly-by and fly-over turn maneuvers, as well as the lateral guidance for tracking a path between two waypoints. The flight plan will consist of the following waypoints: Saint-Hubert Longeuil Airport (CYHU), Pierre-Elliot Trudeau International Airport (CYUL), and Mirabel International Airport (CYMX). Fig. 3.6 shows the interface that the user will use to add waypoints, via the identifier code, and to load the flight plan.



ID	Distance	Track	Flight Level
CYHU	12.797569	96.344864	
CYUL	16.465561	39.602821	
CYMX	---	---	

Code	Lat	Lon	Alt	Freq	Range	Beari...	ID	Airport	Run...	Name
2	0.664...	-1.34...	0		396		APH			A P ...
2	0.996...	0.168...	0		398		GL			AALB...
2	0.982...	0.182...	0		324		ML			AAR...
2	0.982...	0.188...	0		374		TU			AAR...
2	1.199...	-0.92...	0		336		AA			AASI...
2	0.938...	1.595...	0		360		AK			ABA...
2	0.937...	1.594...	0		360		AB			ABA...
2	0.524...	-1.60...	0		230		BNZ			ABBE...

Figure 3.15: Upon adding the three waypoints and pressing the ”Load Flight Plan” button, the distance (nautical miles) and track (degrees East) for each pair of waypoints are computed and displayed for the user.

The aircraft used for the simulator test flights is the Piaggio Avanti, with the model parameters provided by Marinvent. The initial conditions can be seen in Table 3.6. The aircraft is purposely initiated away from the first leg, and with a different heading to that of the course, to emphasize the transient response. Figure 3.6 illustrates the trajectory of the aircraft. The aircraft begins near

Initial Conditions		
Parameter	Value	Units
Latitude	45.4928	degrees N
Longitude	73.4400	degrees E
Altitude	10,000	feet
True Heading	15.66	degrees E
TAS	212	knots

Table 3.1: Initial conditions used for the experimental results.

St. Hubert and turns west to track the leg from CYHU to CYUL. As the aircraft approaches CYUL (marked by the red dot in Figure 3.6), the aircraft performs a fly by turn maneuver and transitions from the first leg (CYHU to CYUL) to the second leg (CYUL to CYMX). Figures 3.6 and 3.6 show the tracking of the desired path.

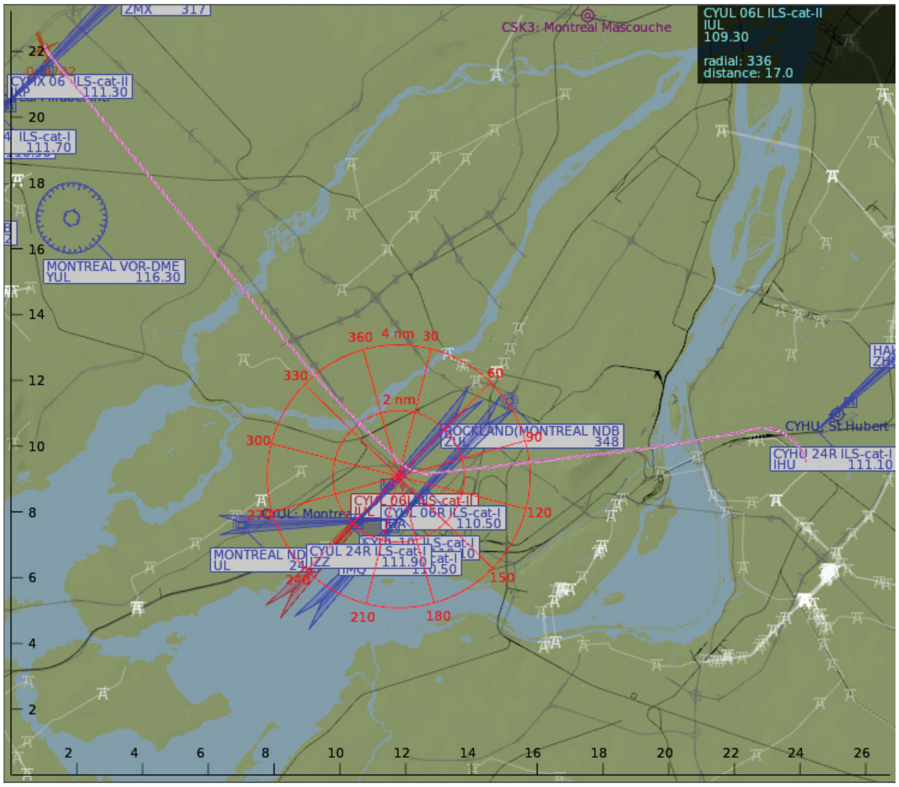


Figure 3.16: The trajectory of the aircraft is denoted by the pink line, from CYHU to CYUL to CYMX.

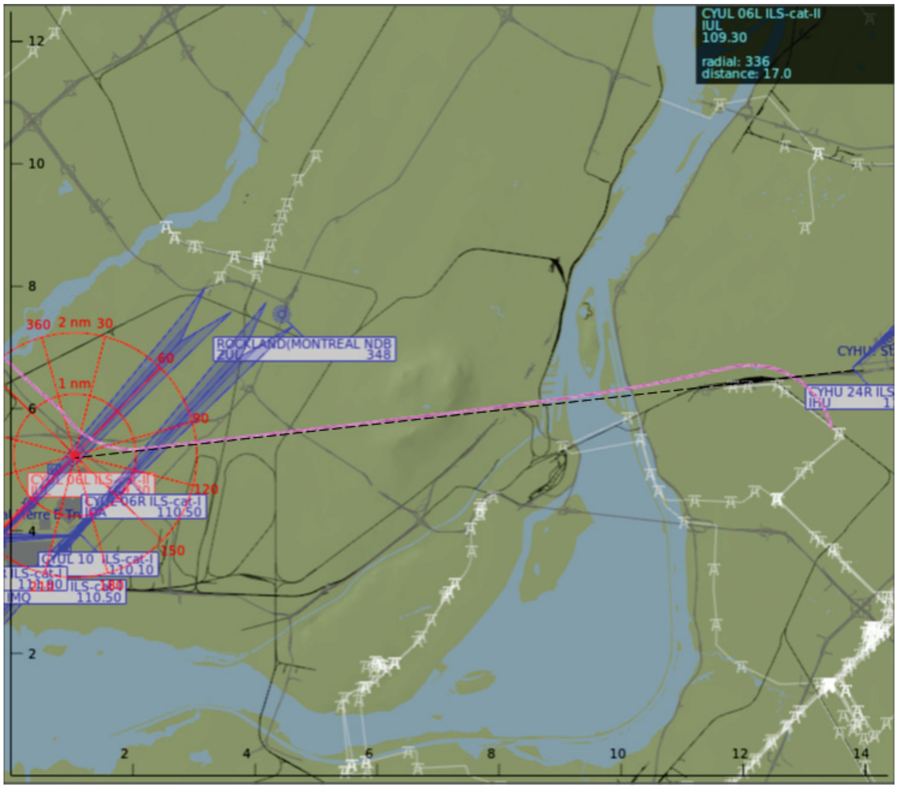


Figure 3.17: The first leg of the flight plan with the dashed line denoting the straight line trajectory between the waypoints.

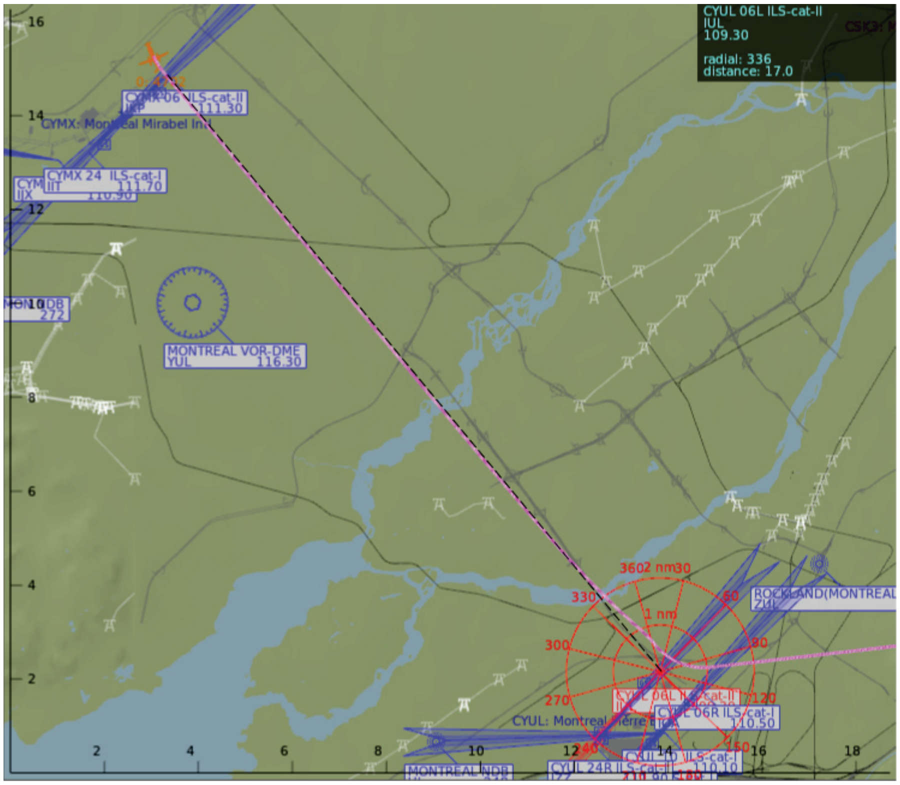


Figure 3.18: The second leg of the flight plan with the dashed line denoting the straight line trajectory between the waypoints.

The results of this section validate the prototype software FMS and meets the proposed functionality stated in section 3.1.

Chapter 4

Multi-agent Coverage Problems and Voronoi Cell Dynamics

This chapter will address multi-agent coverage problems involving agents with second order dynamics. We will consider a partitioned region for which the agents must provide coverage. These problems can be related to many real-life applications such as search and rescue, deployment of mobile network access points, and surveillance. To develop a control law for the agents we will require a property of Voronoi cells, that is derived in this chapter, which is not presently available in the literature to the best of the author's knowledge. This chapter is structured as follows. The preliminaries on Voronoi diagrams and Reynolds Transport Theorem are reviewed in Section 4.1. The dynamics of the area of a Voronoi cell is derived geometrically, and using Reynolds Transport Theorem, in Section 4.2. Section 4.3 derives the dynamics of a Voronoi cell's center of mass using geometry and Reynolds Transport Theorem. Finally, a coverage problem is addressed using the results on Voronoi cells in Section 4.4.

4.1 Preliminaries

This section presents the required background on Voronoi diagrams based on references [29] and [55], and Reynolds Transport Theorem based on [56] and [57].

4.1.1 Voronoi Diagram

Let Q be a bounded convex polytope in $\Re^N$ and $X = (\mathbf{x}_1, ..., \mathbf{x}_n)$ be a set of n distinct points, where the set X is known as the generator set of the Voronoi diagram. Consider the partition of Q into n subsets, such that they only overlap on sets of measure zero. Let the partitioning of Q be the Voronoi diagram $\mathcal{V}(P) = \{V_1, ..., V_n\}$, where $V_i = \{\mathbf{q} \in Q \mid \|\mathbf{q} - \mathbf{x}_i\| \leq \|\mathbf{q} - \mathbf{x}_j\|, \ \forall j \neq i\}$ is defined as the Voronoi cell of generator $\mathbf{x}_i$. Voronoi partitions V_i and V_j are considered neighbouring partitions if $V_i \cap V_j \neq \emptyset$, i.e. the Voronoi partitions share a boundary. Let $\mathcal{N}(i)$ be the set of all generators that are neighbours of generator $\mathbf{x}_i$. Let $\phi : Q \to \Re^+$ be a density function that is measurable and bounded.

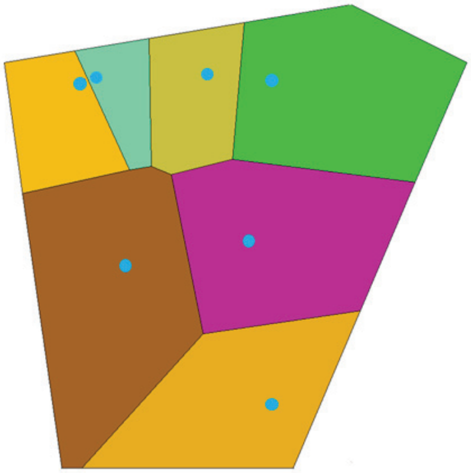


Figure 4.1: Voronoi diagram of a finite convex area.

The mass of Voronoi cell V_i is

$$M_{V_i} = \int_{V_i} \phi(\mathbf{q})\, \mathrm{d}\mathbf{q} \tag{94}$$

and the center of mass, also known as the centroid, is

$$\mathbf{CM}_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} \mathbf{q}\phi(\mathbf{q})\, \mathrm{d}\mathbf{q} \tag{95}$$

4.1.2 Reynolds Transport Theorem

The Reynolds transport theorem (RTT) provides a generalization of the Leibniz rule in $\Re^3$ [58] and has applications in both fluid mechanics and continuum mechanics, among other fields. Consider a non-deforming control volume cv bounded by a control surface cs in a rectangular Cartesian coordinate system. Given the function $\mathcal{A}$, which is continuous with respect to time, (which may yield a scalar, vector, or tensor) and ρ being a continuous function representing the mass density, then

$$\frac{d}{dt}\int_{cv}\mathcal{A}\rho\,\mathrm{d}V = \int_{cv}\frac{\partial}{\partial t}(\mathcal{A}\rho)\,\mathrm{d}V + \int_{cs}\mathcal{A}\rho\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}A \tag{96}$$

where $\mathbf{v}$ is the rate at which the quantity defined by $\mathcal{A}$ crosses the control surface and $\mathbf{n}$ is the outward normal unit vector to the control surface. See [56] for more details on the use of RTT in continuum mechanics. Let us also consider Reynold's transport theorem application to fluid mechanics [57] as follows: Let cv be a control volume, in which the change of properties of the fluid will be observed. Let B represent a property of the fluid (such as kinetic energy) and b represent the fluid property per unit mass (i.e. B/m). The RTT formulation for non-deforming control volumes in fluid mechanics is

$$\frac{dB_{sys}}{dt} = \frac{\partial}{\partial t}\int_{cv}\rho b\,\mathrm{d}V + \int_{cs}\rho b\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}A \tag{97}$$

where ρ is the density of the fluid, cs is the control surface, $\mathbf{v}$ is the velocity of the fluid, and $\mathbf{n}$ is the outwards unit normal vector along the control surface. The result of $\mathbf{v}\cdot\mathbf{n}$ is positive for outflow and negative for inflow of the control volume under consideration. For example, let us consider the conservation of mass law using RTT ($B = m$ and $b = 1$):

$$\dot{m}_{sys} = 0 = \frac{\partial}{\partial t}\int_{cv}\rho\,\mathrm{d}V + \int_{cs}\rho\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}A \tag{98}$$

Since mass can be neither destroyed nor created, equation (98) states that the change in mass in the control volume must be due to the net flow across its control surface.

4.2 Analysis of the Change in Voronoi Cell Area for Moving Generators

This section uses geometry to derive how the area of a Voronoi cell changes with respect to neighbouring generators. The analysis in this section is restricted to 2D Voronoi diagrams. Although a formua for the change in area of a Voronoi cell as a function of the dynamics of its generator is already known (see [30]), this section provides an alternative geometric derivation in 2D that adds considerable insight into the problem. We also provide a formula for the change in center of mass which is new to the best of our knowledge. These results are then proved for general n using Reynolds Transport Theorem.

4.2.1 Geometric Analysis of Change in Area

Let there be n generators in a convex polytope $Q \in \Re^2$ and assume $\phi(\mathbf{q}) = 1, \; \forall \mathbf{q} \in Q$ (which states that the mass of a Voronoi cell is equal to its area). Let $\mathbf{x}_i$ be the generator of Voronoi cell V_i, where $i = 1, ..., n$. Consider the 1D boundary ∂V_{ij} between generators $\mathbf{x}_i$ and $\mathbf{x}_j$, where $j \in \mathcal{N}(i)$. Let $\mathbf{R}_{ij}$ be the vector from generator $\mathbf{x}_i$ to $\mathbf{x}_j$ with norm R_{ij}, $\mathbf{n}_{ij}$ be the unit vector along $\mathbf{R}_{ij}$, and $\mathbf{c}_{ij}$ be the vector from the point of intersection of $\mathbf{R}_{ij}$ and the boundary, to the center of mass of the boundary ∂V_{ij} (see Fig. 4.2). The unit vector along $\mathbf{c}_{ij}$ is defined as $\mathbf{b}_{ij}$. Let the boundary have a finite length, l_{ij}.

Theorem 2. *Given a Voronoi cell V_i, with generator $\boldsymbol{x}_i$, area A_{V_i}, and neighbouring generator $\boldsymbol{x}_j$ where $j \in \mathcal{N}(i)$, the following result holds [30]*

$$\frac{\partial A_{V_i}}{\partial \boldsymbol{x}_j} = -l_{ij}\left(\frac{c_{ij}\boldsymbol{b}_{ij}}{R_{ij}} - \frac{\boldsymbol{n}_{ij}}{2}\right) \tag{99}$$

Proof. Let a local coordinate system be chosen with first axis along the tangential direction to the boundary between cells i and j ($\mathbf{b}_{ij}$) and second axis along the normal to that direction ($\mathbf{n}_{ij}$). Consider a small displacement $\delta\mathbf{x}_j$ of $\mathbf{x}_j$, where $\delta\mathbf{x}_j$ can be decomposed into a component that is parallel to the boundary and a component that is normal to the boundary ∂V_{ij} as follows

$$\delta \mathbf{x}_j = \begin{bmatrix} \delta x_j^1 & \delta x_j^2 \end{bmatrix}^T$$

Fig. 4.2 illustrates the change in the Voronoi cell due to the displacement $\delta \mathbf{x}_j$ which is parallel to the boundary. It is clear from Fig. 4.2 that

$$tan(\phi) = \frac{\delta x_j^1}{R_{ij}} \tag{100}$$

Furthermore, we can partition the area lost and gained by each Voronoi cell as illustrated in Fig. 4.2.

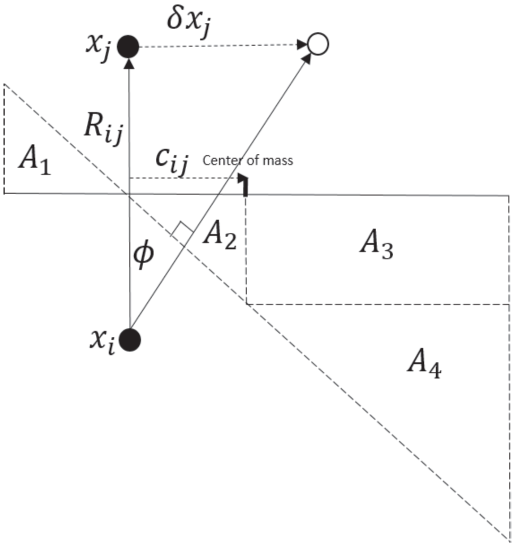


Figure 4.2: Voronoi boundary shift due to a displacment of generator $\mathbf{x}_j$ tangential to the boundary.

The partitioned areas are calculated geometrically as follows.

$$\begin{aligned} A_1 &= \frac{1}{2}\left(\frac{l_{ij}}{2} - c_{ij}\right)^2 \frac{\delta x_j^1}{R_{ij}} \\ A_2 &= \frac{1}{2} c_{ij}^2 \frac{\delta x_j^1}{R_{ij}} \\ A_3 &= \frac{1}{2} l_{ij} c_{ij} \frac{\delta x_j^1}{R_{ij}} \\ A_4 &= \frac{1}{2} \frac{l_{ij}^2}{4} \frac{\delta x_j^1}{R_{ij}} \end{aligned} \tag{101}$$

Using equation (101), the change in area for V_i is calculated as follows.

$$\delta A_{V_i} = A_1 - A_2 - A_3 - A_4 = -\delta x_j^1 l_{ij} \frac{c_{ij}}{R_{ij}} \tag{102}$$

Dividing both sides by δx_j^1, and taking the limit as $\delta x_j^1 \to 0$, yields the first term on the right hand side of equation (99)

$$\frac{\partial A_{V_i}}{\partial x_j^1} = \lim_{\delta x_j^1 \to 0} \frac{\delta A_{V_i}}{\delta x_j^1} = -l_{ij} \frac{c_{ij}}{R_{ij}} \tag{103}$$

Now let us consider a displacement in the direction normal to the boundary, as presented in Fig. 4.3. The intersection point of $\mathbf{R}_{ij}$ and the boundary is $(\mathbf{x}_i + \mathbf{x}_j)/2$, therefore a displacement of δx_j^2 results in a displacement of the boundary by $\frac{1}{2}\delta x_j^2$ as illustrated in Fig. 4.3.

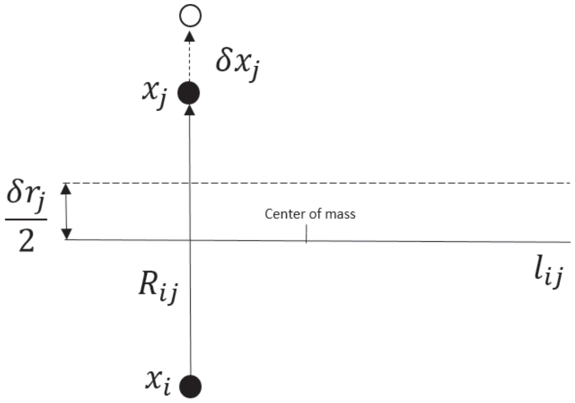


Figure 4.3: Voronoi boundary shift due to a displacment of generator $\mathbf{x}_j$ normal to the boundary.

The change in area is simply

$$\begin{aligned} \delta A_{V_i} &= l_{ij} \frac{\delta x_j^2}{2} \\ \frac{\partial A_{V_i}}{\partial x_j^2} &= \lim_{\delta x_j^2 \to 0} \frac{\delta A_{V_i}}{\delta x_j^2} = \frac{l_{ij}}{2} \end{aligned} \tag{104}$$

which is equivalent to the second term in equation (99). □

One can also analyze how the area of a Voronoi cell changes due to its own generator moving, i.e. $\frac{\partial A_{V_i}}{\partial \mathbf{x}_i}$. The expression can be found in [30] as follows

$$\frac{\partial A_{V_i}}{\partial \mathbf{x}_i} = \sum_{j \in \mathcal{N}(i)} l_{ij} \left(\frac{c_{ij} \mathbf{b}_{ij}}{R_{ij}} + \frac{\mathbf{n}_{ij}}{2} \right) \tag{105}$$

Let us consider a displacement of $\mathbf{x}_i$ by $\delta \mathbf{x}_i$, following the same procedure as described in the proof above. According to Fig. 4.2, if $\mathbf{x}_i$ is displaced instead of $\mathbf{x}_j$ in the direction of $\mathbf{b}_{ij}$, one can clearly see that Voronoi cell V_i will lose area A_1 and gain areas A_2, A_3, and A_4. This can be expressed using equation (103) as follows

$$\frac{\partial A_{V_i}}{\partial x_i^1} = \sum_{j \in \mathcal{N}(i)} - \frac{\partial A_{V_i}}{\partial x_j^1} = \sum_{j \in \mathcal{N}(i)} l_{ij} \frac{c_{ij}}{R_{ij}} \tag{106}$$

Now consider a displacement of $\mathbf{x}_i$ along the normal direction to the boundary as illustrated in Fig. 4.3. One can clearly see that a displacement of $\mathbf{x}_i$ will result in the same displacement of the boundary, and therefore the same change in area as found in (104).

$$\frac{\partial A_{V_i}}{\partial x_i^2} = \sum_{j \in \mathcal{N}(i)} \frac{\partial A_{V_i}}{\partial x_j^2} = \sum_{j \in \mathcal{N}(i)} \frac{l_{ij}}{2} \tag{107}$$

Coalescing the results from (106) and (107) yields the expression in (105).

4.2.2 Reynolds Transport Theorem Analysis of Change in Area

We will now use Reynolds Transport Theorem in the following proof for Theorem 1 to verify our geometric results. This section presents an alternative method by using Reynolds Transport Theorem to determine the change in area. The results are then compared to those of the previous section.

Theorem 3. *The following formula is true for Voronoi cells of dimension n*

$$\left(\frac{\partial \gamma}{\partial x_j^n} \right) \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i) = \frac{1}{2} \boldsymbol{e}_n \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i) - \boldsymbol{e}_n \cdot \left(\gamma - \frac{\boldsymbol{x}_i + \boldsymbol{x}_j}{2} \right) \tag{108}$$

Proof. The perpendicular bisector property of Voronoi diagrams [31] is:

$$\left(\sum_{l\geq 1}\lambda_l\mathbf{u}_l - \frac{\mathbf{x}_i + \mathbf{x}_j}{2}\right)\cdot(\mathbf{x}_j - \mathbf{x}_i) = 0 \tag{109}$$

where l is the integer index of boundary vertices, $\mathbf{u}_l$ is the l^{th} vertex, and $\lambda_l \in [0,1]$, where $\sum \lambda_l = 1$. The sum of $\lambda_l\mathbf{u}_l$ is the convex combination representing vectors on the boundary (see Fig. 4.4).

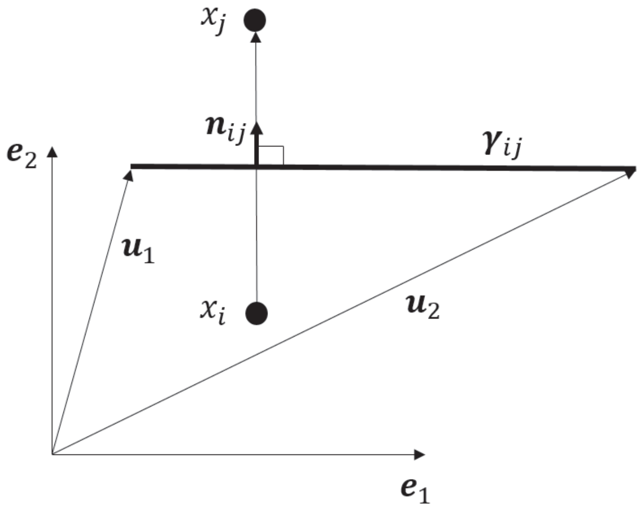


Figure 4.4: A convex combination of the boundary vertices to represent the boundary between x_i and x_j.

Let $\mathbf{e}_n$ be the unit vector in the n^{th} direction of space, x_i^n be the n^{th} component of the vector $\mathbf{x}_i$, and $\boldsymbol{\gamma} = \sum \lambda_l\mathbf{u}_l$, which is a parameterization of the boundary ∂V_{ij}. Differentiating equation (109) relative to x_j^n yields [31]

$$\left(\frac{\partial\gamma}{\partial x_j^n} - \frac{1}{2}\mathbf{e}_n\right)\cdot(\mathbf{x}_j - \mathbf{x}_i) + \left(\gamma - \frac{\mathbf{x}_i + \mathbf{x}_j}{2}\right)\cdot\mathbf{e}_n = 0 \tag{110}$$

Expanding and re-arranging (110) yields

$$\left(\frac{\partial\gamma}{\partial x_j^n}\right)\cdot(\mathbf{x}_j - \mathbf{x}_i) = \frac{1}{2}\mathbf{e}_n\cdot(\mathbf{x}_j - \mathbf{x}_i) - \mathbf{e}_n\cdot\left(\gamma - \frac{\mathbf{x}_i + \mathbf{x}_j}{2}\right)$$

□

Let us now consider another proof for the result in Theorem 2.

Proof. The area of a Voronoi cell can be expressed as follows.

$$A_{V_i} = \int_{V_i} 1 \,\mathrm{d}\mathbf{q} \tag{111}$$

Taking the partial derivative with respect to $\mathbf{x}_j$ of equation (111) and applying Reynolds transport theorem [57] yields

$$\frac{\partial A_{V_i}}{\partial \mathbf{x}_j} = \frac{\partial}{\partial \mathbf{x}_j} \int_{V_i} 1 \,\mathrm{d}\mathbf{q} = \int_{\partial V_i} \frac{\partial \boldsymbol{\gamma}}{\partial \mathbf{x}_j} \mathbf{n}(\boldsymbol{\gamma}) \,\mathrm{d}\gamma^1 \tag{112}$$

where the coordinate system is such that γ^1 is the component of the boundary parameterization $\boldsymbol{\gamma}$ tangent to the boundary. Comparing equation (112) to (96), we have considered $\mathcal{A} = 1$, $\rho = 1$, and $\mathbf{v} = \frac{\partial \boldsymbol{\gamma}}{\partial \mathbf{x}_j}$ is the velocity of the boundary. Note that we can consider the partial derivative of A_{V_i} with respect to $\mathbf{x}_j$ to be equivalent to its total derivative with respect to $\mathbf{x}_j$ since the area of the Voronoi cell does not vary according to other variables, hence it can be treated as holding all other variables constant. Let us split equation (112) into two components

$$\begin{aligned} \frac{\partial}{\partial x_j^1} \int_{V_i} 1 \,\mathrm{d}\mathbf{q} &= \int_{\partial V_i} \frac{\partial \boldsymbol{\gamma}}{\partial x_j^1} \cdot \mathbf{n}(\boldsymbol{\gamma}) \,\mathrm{d}\gamma^1 \\ \frac{\partial}{\partial x_j^2} \int_{V_i} 1 \,\mathrm{d}\mathbf{q} &= \int_{\partial V_i} \frac{\partial \boldsymbol{\gamma}}{\partial x_j^2} \cdot \mathbf{n}(\boldsymbol{\gamma}) \,\mathrm{d}\gamma^1 \end{aligned} \tag{113}$$

Now we will use the results presented in equation (110), divide by the norm R_{ij}, and substitute it into equation (112) which yields

$$\begin{aligned} \frac{\partial A_{V_i}}{\partial x_j^1} &= \int_{\partial V_i} \frac{1}{2R_{ij}} \mathbf{e}_1 \cdot (\mathbf{x}_j - \mathbf{x}_i) - \frac{\mathbf{e}_1}{R_{ij}} \cdot \left(\boldsymbol{\gamma} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2} \right) \mathrm{d}\gamma^1 \\ \frac{\partial A_{V_i}}{\partial x_j^2} &= \int_{\partial V_i} \frac{1}{2R_{ij}} \mathbf{e}_2 \cdot (\mathbf{x}_j - \mathbf{x}_i) - \frac{\mathbf{e}_2}{R_{ij}} \cdot \left(\boldsymbol{\gamma} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2} \right) \mathrm{d}\gamma^1 \end{aligned} \tag{114}$$

Expanding equation (114) yields

$$\begin{aligned} \frac{\partial A_{V_i}}{\partial x_j^1} &= \int_{\partial V_i} \frac{1}{2R_{ij}} \mathbf{e}_1 \cdot (\mathbf{x}_j - \mathbf{x}_i) - \frac{\mathbf{e}_1}{R_{ij}} \cdot \boldsymbol{\gamma} + \frac{\mathbf{e}_1}{R_{ij}} \cdot \left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2} \right) \mathrm{d}\gamma^1 \\ \frac{\partial A_{V_i}}{\partial x_j^2} &= \int_{\partial V_i} \frac{1}{2R_{ij}} \mathbf{e}_2 \cdot (\mathbf{x}_j - \mathbf{x}_i) - \frac{\mathbf{e}_2}{R_{ij}} \cdot \boldsymbol{\gamma} + \frac{\mathbf{e}_2}{R_{ij}} \cdot \left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2} \right) \mathrm{d}\gamma^1 \end{aligned} \tag{115}$$

Let us consider the coordinate system described by unit vectors $\mathbf{b}_{ij}$ and $\mathbf{n}_{ij}$, as defined in section 4.2.1, where the origin of the coordinate system lies on the boundary at the midpoint between the two generators $\mathbf{x}_j$ and $\mathbf{x}_i$. We can now simplify equation (115) using the following geometric properties,

$$\begin{aligned} \mathbf{b}_{ij} \cdot (\mathbf{x}_j - \mathbf{x}_i) &= 0 \\ \frac{\mathbf{x}_i + \mathbf{x}_j}{2} &= \mathbf{0} \\ \mathbf{n}_{ij} \cdot \boldsymbol{\gamma} &= 0 \end{aligned} \tag{116}$$

where $\mathbf{e}_1$ has been replaced by $\mathbf{b}_{ij}$ and $\mathbf{e}_2$ has been replaced by $\mathbf{n}_{ij}$, which yields

$$\begin{aligned} \frac{\partial A_{V_i}}{\partial x_j^1} &= -\frac{1}{R_{ij}} \int_{\partial V_i} \gamma^1 \, \mathrm{d}\gamma^1 \\ \frac{\partial A_{V_i}}{\partial x_j^2} &= \int_{\partial V_i} \frac{1}{2} \, \mathrm{d}\gamma^1 \end{aligned} \tag{117}$$

Evaluating the integral along the boundary from $-\frac{l_{ij}}{2} + c_{ij}$ to $\frac{l_{ij}}{2} + c_{ij}$ yields

$$\frac{\partial A_{V_i}}{\partial \mathbf{x}_j} = -l_{ij} \left(\frac{c_{ij}}{R_{ij}} \mathbf{b}_{ij} - \frac{1}{2} \mathbf{n}_{ij} \right) \tag{118}$$

which is the result obtained in [30] and the geometric approach presented in section 4.2.1. □

Now let us consider the change in area of Voronoi cell V_i due to generator $\mathbf{x}_i$

$$\frac{\partial A_{V_i}}{\partial \mathbf{x}_i} = \frac{\partial}{\partial \mathbf{x}_i} \int_{V_i} 1 \, \mathrm{d}\mathbf{q} = \sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \frac{\partial \boldsymbol{\gamma}}{\partial \mathbf{x}_i} \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 \tag{119}$$

Following the same procedure as shown in the proof above, the partial derivative of (109) with respect to $\mathbf{x}_i$ yields

$$\left(\frac{\partial \boldsymbol{\gamma}}{\partial x_i^n} - \frac{1}{2} \mathbf{e}_n \right) \cdot (\mathbf{x}_j - \mathbf{x}_i) - \left(\boldsymbol{\gamma} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2} \right) \cdot \mathbf{e}_n = 0 \tag{120}$$

where $\boldsymbol{\gamma} = \sum \lambda_l \mathbf{u}_l$. Expanding and re-arranging (120) yields

$$\left(\frac{\partial \boldsymbol{\gamma}}{\partial x_i^n} \right) \cdot (\mathbf{x}_j - \mathbf{x}_i) = \frac{1}{2} \mathbf{e}_n \cdot (\mathbf{x}_j - \mathbf{x}_i) + \mathbf{e}_n \cdot \left(\boldsymbol{\gamma} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2} \right) \tag{121}$$

Dividing (120) by R_{ij} and substituting it into equation (119) yields

$$\begin{aligned}\frac{\partial A_{V_i}}{\partial x_i^1} &= \sum_{j\in\mathcal{N}(i)} \int_{\partial V_{ij}} \frac{1}{2R_{ij}}\mathbf{e}_1\cdot(\mathbf{x}_j-\mathbf{x}_i)+\frac{\mathbf{e}_1}{R_{ij}}\cdot\left(\boldsymbol{\gamma}-\frac{\mathbf{x}_i+\mathbf{x}_j}{2}\right)\mathrm{d}\gamma^1 \\ \frac{\partial A_{V_i}}{\partial x_i^2} &= \sum_{j\in\mathcal{N}(i)} \int_{\partial V_{ij}} \frac{1}{2R_{ij}}\mathbf{e}_2\cdot(\mathbf{x}_j-\mathbf{x}_i)+\frac{\mathbf{e}_2}{R_{ij}}\cdot\left(\boldsymbol{\gamma}-\frac{\mathbf{x}_i+\mathbf{x}_j}{2}\right)\mathrm{d}\gamma^1\end{aligned} \tag{122}$$

where ∂V_{ij} is the boundary formed between Voronoi cells V_i and V_j. Expanding equation (122) yields

$$\begin{aligned}\frac{\partial A_{V_i}}{\partial x_i^1} &= \sum_{j\in\mathcal{N}(i)} \int_{\partial V_{ij}} \frac{1}{2R_{ij}}\mathbf{e}_1\cdot(\mathbf{x}_j-\mathbf{x}_i)+\frac{\mathbf{e}_1}{R_{ij}}\cdot\boldsymbol{\gamma}-\frac{\mathbf{e}_1}{R_{ij}}\cdot\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}\right)\mathrm{d}\gamma^1 \\ \frac{\partial A_{V_i}}{\partial x_i^2} &= \sum_{j\in\mathcal{N}(i)} \int_{\partial V_{ij}} \frac{1}{2R_{ij}}\mathbf{e}_2\cdot(\mathbf{x}_j-\mathbf{x}_i)+\frac{\mathbf{e}_2}{R_{ij}}\cdot\boldsymbol{\gamma}-\frac{\mathbf{e}_2}{R_{ij}}\cdot\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}\right)\mathrm{d}\gamma^1\end{aligned} \tag{123}$$

Simplifying equation (123) according to the geometric properties in (116), where $\mathbf{e}_1$ has been replaced by $\mathbf{b}_{ij}$ and $\mathbf{e}_2$ has been replaced by $\mathbf{n}_{ij}$, yields

$$\begin{aligned}\frac{\partial A_{V_i}}{\partial x_i^1} &= \sum_{j\in\mathcal{N}(i)} \frac{1}{R_{ij}}\int_{\partial V_{ij}} \gamma^1\,\mathrm{d}\gamma^1 \\ \frac{\partial A_{V_i}}{\partial x_i^2} &= \sum_{j\in\mathcal{N}(i)} \int_{\partial V_{ij}} \frac{1}{2}\,\mathrm{d}\gamma^1\end{aligned} \tag{124}$$

Evaluating the integral along the boundary from $-\frac{l_{ij}}{2}+c_{ij}$ to $\frac{l_{ij}}{2}+c_{ij}$ yields

$$\frac{\partial A_{V_i}}{\partial \mathbf{x}_i} = \sum_{j\in\mathcal{N}(i)} l_{ij}\left(\frac{c_{ij}}{R_{ij}}\mathbf{b}_{ij}+\frac{1}{2}\mathbf{n}_{ij}\right) \tag{125}$$

Equation (125) matches the geometric result in (105) and the work in [30].

4.3 Analysis of the Change in the Center of Mass of a Voronoi Cell for Moving Generators

The center of mass of a Voronoi cell is of great importance to coverage problems, where the optimal coverage is obtained by having agents (generators) located at the center of mass of their respective Voronoi cell [59]. This configuration is defined as a centroidal Voronoi tessellation (CVT).

Lloyd's algorithm [60] [31] provides a method to find CVTs. The algorithm is performed iteratively, by computing the Voronoi diagram and moving each generator towards the center of mass of their respective Voronoi cell. Knowledge of the movement of the center of mass of Voronoi cells may prove useful in future research into coverage problems, for example the movement of the center of mass of Voronoi cells will be needed in section 4.4 of this thesis. This section will present the change in position of the center of mass of a Voronoi cell using both geometric analysis and Reynolds Transport Theorem.

4.3.1 Geometric Analysis of the Change in Center of Mass

This section will apply geometric techniques to analyze the change in position of the center of mass of a given Voronoi cell due to displacements of neighbouring generators in 2D.

Theorem 4. *Given a Voronoi cell V_i, with area A_{V_i}, center of mass $\boldsymbol{CM}_{V_i}$ having vector components $CM^1_{V_i}$ and $CM^2_{V_i}$, and neighbouring generator $\boldsymbol{x}_j$ where $j \in \mathcal{N}(i)$, the following result holds:*

$$\frac{\partial \boldsymbol{CM}_{V_i}}{\partial \boldsymbol{x}_j} = \frac{l_{ij}}{A_{V_i}} \begin{bmatrix} \frac{1}{R_{ij}}(c_{ij} CM^1_{V_i} - \frac{l^2_{ij}}{12} - c^2_{ij}) & \frac{1}{2}(c_{ij} - CM^1_{V_i}) \\ \frac{c_{ij}}{R_{ij}} CM^2_{V_i} & -\frac{1}{2} CM^2_{V_i} \end{bmatrix} \tag{126}$$

Proof. The same local coordinate system used in the proof of the change in area will be used here. Consider the displacement of the boundary illustrated in Figure 4.2 due to a displacement of the neighbouring generator along the parallel direction to the boundary. Assume a uniform density of 1 over the Voronoi diagram (i.e. $\phi(\mathbf{q}) = 1, \forall \mathbf{q} \in Q$). One can substitute the mass of the Voronoi cells for its area. The change in the position of the center of mass of V_i is then

$$\delta \mathbf{CM}_{V_i} = \mathbf{CM}_{V_i + \delta V_i} - \mathbf{CM}_{V_i} \tag{127}$$

where $\mathbf{CM}_{V_i+\delta V_i}$ is the new center of mass after the boundary displacement, which can be expressed as

$$\mathbf{CM}_{V_i+\delta V_i} = \frac{\mathbf{CM}_{V_i} A_{V_i} + \mathbf{CM}_{\delta V_i} \delta A_{V_i}}{A_{V_i} + \delta A_{V_i}} \tag{128}$$

where $\mathbf{CM}_{\delta V_i}$ is the center of mass of the change in area and δA_{V_i} is defined in equation (102).

Substituting (128) into (127) and simplifying yields

$$\delta\mathbf{CM}_{V_i} = \frac{\mathbf{CM}_{\delta V_i}\delta A_{V_i} - \mathbf{CM}_{V_i}\delta A_{V_i}}{A_{V_i} + \delta A_{V_i}} \tag{129}$$

where for displacements in the direction parallel to the boundary $\mathbf{CM}_{\delta V_i}$ can be expressed as

$$\begin{aligned} \mathbf{CM}_{\delta V_i} &= \frac{A_1\mathbf{CM}_{A_1} - A_2\mathbf{CM}_{A_2} - A_3\mathbf{CM}_{A_3} - A_4\mathbf{CM}_{A_4}}{\delta A_{V_i}} \\ &= \frac{A_1\mathbf{CM}_{A_1} - A_2\mathbf{CM}_{A_2} - A_3\mathbf{CM}_{A_3} - A_4\mathbf{CM}_{A_4}}{A_1 - A_2 - A_3 - A_4} \end{aligned} \tag{130}$$

Using the geometry of Fig. 4.2, the center of mass for each area is:

$$\begin{aligned} \mathbf{CM}_{A_1} &= \frac{1}{3}(-l_{ij} + 2c_{ij})\mathbf{b}_{ij} + \frac{1}{3}\frac{\delta x_j^1}{R_{ij}}(\frac{l_{ij}}{2} - c_{ij})\mathbf{n}_{ij} \\ \mathbf{CM}_{A_2} &= \frac{1}{3}2c_{ij}\mathbf{b}_{ij} - \frac{1}{3}\frac{\delta x_j^1}{R_{ij}}c_{ij}\mathbf{n}_{ij} \\ \mathbf{CM}_{A_3} &= (\frac{l_{ij}}{4} + c_{ij})\mathbf{b}_{ij} - \frac{1}{2}\frac{\delta x_j^1}{R_{ij}}c_{ij}\mathbf{n}_{ij} \\ \mathbf{CM}_{A_4} &= \frac{1}{3}(l_{ij} + 3c_{ij})\mathbf{b}_{ij} - \frac{1}{3}\frac{\delta x_j^1}{R_{ij}}(3c_{ij} + \frac{1}{2}l_{ij})\mathbf{n}_{ij} \end{aligned} \tag{131}$$

Substituting equations (131), (101), and (130) into equation (129) yields:

$$\frac{\partial\mathbf{CM}_{V_i}}{\partial x_j^1} = \lim_{\delta x_j^1 \to 0}\frac{\delta\mathbf{CM}_{V_i}}{\delta x_j^1} = \frac{l_{ij}}{R_{ij}A_{V_i}}\left(c_{ij}\mathbf{CM}_{V_i} - (\frac{1}{12}l_{ij}^2 + c_{ij}^2)\mathbf{b}_{ij}\right) \tag{132}$$

If one now considers a displacement of generator $\mathbf{x}_j$ along the normal direction of the boundary (see Fig. 4.3) the result from equation (129) can be used to yield

$$\delta\mathbf{CM}_{V_i} = \frac{\delta x_j^2 l_{ij}}{2}\left(\begin{bmatrix} c_{ij} \\ \frac{\delta x_j^2}{4} \end{bmatrix} - \mathbf{CM}_{V_i}\right)\frac{1}{A_{V_i} + \frac{\delta x_j^2}{2}l_{ij}} \tag{133}$$

Dividing both sides of equation (133) by δx_j^2 and taking the limit as δx_j^2 approaches 0 yields

$$\begin{aligned}\frac{\partial \mathbf{CM}_{V_i}}{\partial x_j^2} = \lim_{\delta x_j^2 \to 0} \frac{\delta \mathbf{CM}_{V_i}}{\delta x_j^2} &= \frac{l_{ij}}{2A_{V_i}} \left(\begin{bmatrix} c_{ij} \\ 0 \end{bmatrix} - \mathbf{CM}_{V_i} \right) \\ &= \frac{l_{ij}}{2A_{V_i}} \left(c_{ij} \mathbf{b}_{ij} - \mathbf{CM}_{V_i} \right)\end{aligned} \tag{134}$$

Coalescing the results in equations (132) and (134) yields the following Jacobian matrix

$$\frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_j} = \frac{l_{ij}}{A_{V_i}} \begin{bmatrix} \frac{1}{R_{ij}}(c_{ij} CM^1_{V_i} - \frac{l_{ij}^2}{12} - c_{ij}^2) & \frac{1}{2}(c_{ij} - CM^1_{V_i}) \\ \frac{c_{ij}}{R_{ij}} CM^2_{V_i} & -\frac{1}{2} CM^2_{V_i} \end{bmatrix}$$

□

4.3.2 Reynolds Transport Theorem Analysis of Change in Center of Mass

The following section presents an alternate proof to Theorem 4 by applying Reynolds Transport Theorem.

Proof. The center of mass of Voronoi cell V_i, using uniform density, is

$$\mathbf{CM}_{V_i} = \frac{1}{A_{V_i}} \int_{V_i} \mathbf{q} \, \mathrm{d}\mathbf{q} \tag{135}$$

Taking the partial derivative of equation (135) considered as a product of two terms, applying RTT to the second term, and taking into account that the dynamics of the agents only affect the position of the boundary between cells yields

$$\frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_j} = -\frac{1}{A_{V_i}} \mathbf{CM}_{V_i} \left(\frac{\partial A_{V_i}}{\partial \mathbf{x}_j} \right)^T + \frac{1}{A_{V_i}} \int_{\partial V_i} \gamma \frac{\partial \gamma}{\partial \mathbf{x}_j} \mathbf{n}(\gamma) \, \mathrm{d}\gamma^1 \tag{136}$$

where, as before, γ^1 is the component of the boundary parameterization along the tangent direction to the boundary (i.e, the direction of $\mathbf{b}_{ij}$). Splitting the integral into two components, and using the result of equation (108) divided by R_{ij} yields

$$\begin{aligned}
&\int_{\partial V_i} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_j^1} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 = \\
&\int_{\partial V_i} \boldsymbol{\gamma} \left(\frac{1}{2} \mathbf{b}_{ij} \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{R_{ij}} - \frac{\mathbf{b}_{ij}}{R_{ij}} \cdot \left(\boldsymbol{\gamma} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2} \right) \right) \mathrm{d}\gamma^1 \\
&\int_{\partial V_i} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_j^2} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 = \\
&\int_{\partial V_i} \boldsymbol{\gamma} \left(\frac{1}{2} \mathbf{n}_{ij} \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{R_{ij}} - \frac{\mathbf{n}_{ij}}{R_{ij}} \cdot \left(\boldsymbol{\gamma} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2} \right) \right) \mathrm{d}\gamma^1
\end{aligned} \tag{137}$$

Applying the properties in equation (116), we can further simplify equation (137) to

$$\begin{aligned}
\int_{\partial V_i} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_j^1} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 &= -\int_{\partial V_i} \frac{1}{R_{ij}} \boldsymbol{\gamma} \gamma^1 \, \mathrm{d}\gamma^1 \\
\int_{\partial V_i} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_j^2} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 &= \int_{\partial V_i} \frac{1}{2} \boldsymbol{\gamma} \, \mathrm{d}\gamma^1
\end{aligned} \tag{138}$$

Substituting $\boldsymbol{\gamma} = \gamma^1 \mathbf{b}_{ij}$ into (138) and integrating between the limits $[-\frac{l_{ij}}{2} + c_{ij}, \frac{l_{ij}}{2} + c_{ij}]$ yields

$$\int_{\partial V_i} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_j^1} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 \quad = \quad -\left[\frac{(\gamma^1)^3}{3R_{ij}} \right]_{-\frac{l_{ij}}{2}+c_{ij}}^{\frac{l_{ij}}{2}+c_{ij}} \mathbf{b}_{ij} \tag{139}$$

$$\int_{\partial V_i} \frac{1}{2} \gamma^1 \mathbf{b}_{ij} \, \mathrm{d}\gamma^1 \quad = \quad \left[\frac{(\gamma^1)^2}{4} \right]_{-\frac{l_{ij}}{2}+c_{ij}}^{\frac{l_{ij}}{2}+c_{ij}} \mathbf{b}_{ij} \tag{140}$$

Evaluating this result yields

$$\begin{aligned}
\int_{\partial V_i} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_j^1} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 &= -\frac{l_{ij}}{R_{ij}} \begin{bmatrix} \left(\frac{l_{ij}^2}{12} + c_{ij}^2 \right) \\ 0 \end{bmatrix} \\
\int_{\partial V_i} \frac{1}{2} \gamma^1 \mathbf{b}_{ij} \, \mathrm{d}\gamma^1 &= \begin{bmatrix} \frac{l_{ij} c_{ij}}{2} \\ 0 \end{bmatrix}
\end{aligned} \tag{141}$$

Substituting equation (118) and (141) into equation (136)

$$\frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_j} = \frac{l_{ij}}{A_{V_i}} \mathbf{CM}_{V_i} \begin{bmatrix} \frac{c_{ij}}{R_{ij}} & -\frac{1}{2} \end{bmatrix} + \frac{l_{ij}}{A_{V_i}} \begin{bmatrix} \frac{1}{R_{ij}} \left(-\frac{l_{ij}^2}{12} - c_{ij}^2 \right) & \frac{c_{ij}}{2} \\ 0 & 0 \end{bmatrix} \tag{142}$$

and then expanding yields

$$\frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_j} = \frac{l_{ij}}{A_{V_i}} \begin{bmatrix} \frac{1}{R_{ij}}\left(c_{ij}CM^1_{V_i} - \frac{l_{ij}^2}{12} - c_{ij}^2\right) & \frac{1}{2}\left(c_{ij} - CM^1_{V_i}\right) \\ \frac{c_{ij}}{R_{ij}}CM^2_{V_i} & -\frac{1}{2}CM^2_{V_i} \end{bmatrix} \tag{143}$$

Note that equation (143) matches the results that were determined geometrically in equation (126).

□

Using the same procedure as shown in the proof above, one can determine the change in the position of the center of mass $\mathbf{CM}_{V_i}$ with respect to its own Voronoi cell generator $\mathbf{x}_i$ as follows.

$$\frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_i} = -\frac{1}{A_{V_i}}\mathbf{CM}_{V_i}\left(\frac{\partial A_{V_i}}{\partial \mathbf{x}_i}\right)^T + \frac{1}{A_{V_i}} \sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial \mathbf{x}_i} \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 \tag{144}$$

where ∂V_{ij} is the boundary between Voronoi cells V_i and V_j. Splitting the integral into two components, and dividing equation (121) by R_{ij} yields

$$\begin{aligned} &\sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_i^1} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 = \\ &\sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \boldsymbol{\gamma} \left(\frac{1}{2}\mathbf{b}_{ij} \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{R_{ij}} + \frac{\mathbf{b}_{ij}}{R_{ij}} \cdot \left(\boldsymbol{\gamma} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2}\right)\right) \mathrm{d}\gamma^1 \\ &\sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_i^2} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 = \\ &\sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \boldsymbol{\gamma} \left(\frac{1}{2}\mathbf{n}_{ij} \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{R_{ij}} + \frac{\mathbf{n}_{ij}}{R_{ij}} \cdot \left(\boldsymbol{\gamma} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2}\right)\right) \mathrm{d}\gamma^1 \end{aligned} \tag{145}$$

Applying the properties in equation (116), we can further simplify equation (145) to

$$\begin{aligned} \sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_i^1} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 &= \sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \frac{1}{R_{ij}} \boldsymbol{\gamma}\gamma^1 \, \mathrm{d}\gamma^1 \\ \sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_i^2} \cdot \mathbf{n}(\boldsymbol{\gamma}) \, \mathrm{d}\gamma^1 &= \int_{\partial V_{ij}} \frac{1}{2} \boldsymbol{\gamma} \, \mathrm{d}\gamma^1 \end{aligned} \tag{146}$$

Substituting $\boldsymbol{\gamma} = \gamma^1 \mathbf{b}_{ij}$ into (138) and integrating between the limits $[-\frac{l_{ij}}{2} + c_{ij}, \frac{l_{ij}}{2} + c_{ij}]$ yields

$$\begin{aligned} \sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \boldsymbol{\gamma} \frac{\partial \boldsymbol{\gamma}}{\partial x_i^1} \cdot \mathbf{n}(\boldsymbol{\gamma})\, \mathrm{d}\gamma^1 &= \sum_{j \in \mathcal{N}(i)} \frac{l_{ij}}{R_{ij}} \begin{bmatrix} \left(\frac{l_{ij}^2}{12} + c_{ij}^2 \right) \\ 0 \end{bmatrix} \\ \sum_{j \in \mathcal{N}(i)} \int_{\partial V_{ij}} \frac{1}{2} \gamma^1 \mathbf{b}_{ij}\, \mathrm{d}\gamma^1 &= \begin{bmatrix} \frac{l_{ij} c_{ij}}{2} \\ 0 \end{bmatrix} \end{aligned} \tag{147}$$

Substituting equation (125) and (147) into equation (144)

$$\frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_i} = \sum_{j \in \mathcal{N}(i)} \frac{l_{ij}}{A_{V_i}} \mathbf{CM}_{V_i} \begin{bmatrix} \frac{c_{ij}}{R_{ij}} & \frac{1}{2} \end{bmatrix} + \frac{l_{ij}}{A_{V_i}} \begin{bmatrix} \frac{1}{R_{ij}} \left(\frac{l_{ij}^2}{12} + c_{ij}^2 \right) & \frac{c_{ij}}{2} \\ 0 & 0 \end{bmatrix} \tag{148}$$

and then expanding yields

$$\frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_i} = \sum_{j \in \mathcal{N}(i)} \frac{l_{ij}}{A_{V_i}} \begin{bmatrix} \frac{1}{R_{ij}} \left(c_{ij} CM_{V_i}^1 + \frac{l_{ij}^2}{12} + c_{ij}^2 \right) & \frac{1}{2} \left(c_{ij} + CM_{V_i}^1 \right) \\ \frac{c_{ij}}{R_{ij}} CM_{V_i}^2 & \frac{1}{2} CM_{V_i}^2 \end{bmatrix} \tag{149}$$

4.4 Voronoi Dynamics in Multi-agent Coverage Problems

The following section will illustrate the applications of Voronoi cell dynamics to multi-agent coverage problems.

4.4.1 Coverage Problem Preliminaries

This section will be focused on reviewing the work presented in [59] and [34]. Let there be n agents in a region of space in $\Re^2$ and assume that each agent, $i = 1, .., n$, has dynamics of the form

$$\dot{\mathbf{x}}_i = \mathbf{u}_i \tag{150}$$

where $\mathbf{x}_i$ is the position of agent i and $\mathbf{u}_i$ is the corresponding control input. Let Q be a convex polytope in $\Re^2$ and $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_n]^T$ be the vector of the locations of the agents. Consider the partition of Q into i subsets, such that they only overlap on sets of measure zero. Let the agent's positions **x** be the generators for a Voronoi diagram, where the Voronoi diagram is defined in section 4.1.1. One aims at providing the ”best” or ”optimal” coverage when solving a coverage problem for a given region. Let the function $f(\|\mathbf{x}_i - \mathbf{q}\|) = \|\mathbf{x}_i - \mathbf{q}\|^2$ represent how poor the coverage is for a given point $\mathbf{q} \in Q$ [59] (i.e. the smaller the value of $f(\|\mathbf{x}_i - \mathbf{q}\|)$, the better the coverage). Consider the density function $\phi(\mathbf{q})$, as defined in section 4.1.1, which provides a weight on the importance of the coverage at the point $\mathbf{q}$. Let each agent be responsible for the coverage in its corresponding Voronoi partition. Consider the candidate Lyapunov function

$$V(\mathbf{x}) = \sum_{i=1}^{n} \int_{V_i} ||\mathbf{x}_i - \mathbf{q}||^2 \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} \tag{151}$$

Taking the partial derivative with respect to the agent’s position $\mathbf{x}_i$ yields [59]

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_i} V(\mathbf{x}) &= 2 \int_{V_i} (\mathbf{x}_i - \mathbf{q}) \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} \\ &= 2 \bigg(\int_{V_i} \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} \bigg) \bigg(\mathbf{x}_i - \frac{\int_{V_i} \mathbf{q} \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q}}{\int_{V_i} \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q}} \bigg) \end{aligned} \tag{152}$$

Substituting (94) and (95) into (152) yields

$$\frac{\partial}{\partial \mathbf{x}_i} V(\mathbf{x}) = 2 M_{V_i} (\mathbf{x}_i - \mathbf{CM}_{V_i}) \tag{153}$$

and taking the second partial derivative yields

$$\frac{\partial^2}{\partial \mathbf{x}_i^2} V(\mathbf{x}) = 2 M_{V_i} > 0 \tag{154}$$

Equations (153) and (154) demonstrate that a local minimum is achieved when $\mathbf{x}_i = \mathbf{CM}_{V_i}$, this configuration is known as the Centroidal Voronoi Configuration (CVC).

4.4.2 Energy-Efficient Coverage Optimal Control Problem

The work in [34] formulates the coverage problem as an optimal control problem. The optimal coverage control problem is described by the following cost function

$$J = \int_0^\infty \sum_{i=1}^n s_i || \int_{V_i} (\mathbf{x}_i - \mathbf{q})\phi(\mathbf{q}) \, \mathrm{d}\mathbf{q}||^2 + r_i \mathbf{u}_i^T \mathbf{u}_i \, \mathrm{d}\tau \tag{155}$$

where n is the number of agents, $s_i \geq 0$, $r_i > 0$, V_i is the Voronoi cell corresponding to the i^{th} agent, and $\mathbf{x}_i$ the position of the i^{th} agent. Reference [34] proves that the optimal control input that minimizes the cost function in (155) subject to the dynamics in (150) and is spatially distributed over Delaunay graphs is

$$\mathbf{u}_i = -\sqrt{\frac{s_i}{r_i}} \int_{V_i} (\mathbf{x}_i - \mathbf{q})\phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} \tag{156}$$

If $s_i r_i = 1, \forall i \in \{1, ..., n\}$ then it is proven in [34] that the value function (which is also a Lyapunov function) is

$$V(\mathbf{x}) = \sum_{i=1}^n \int_{V_i} ||(\mathbf{x}_i - \mathbf{q})||^2 \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} \tag{157}$$

which is the same as the one in (151). The system therefore converges to a Centroidal Voronoi Configuration. It is important to note that the optimal control input only depends on each agent being able to compute its own Voronoi cell, which means the control input for each agent is only dependent on its own position and the position of its neighbouring agents. This allows for a scalable solution in practice as the number of agents increase. It is important to note that although the coverage criterion being minimized in (155) differs from (151), the value function that is obtained as the spatially distributed solution to the optimal control problem coincides with (151). This offers an important additional interpretation of the function (151).

4.4.3 Coverage Problem with Second Order Dynamics

We now use the velocity as a virtual control input that should be steered to the optimal value of the velocity obtained in the previous section. It will also be assumed that the density function is uniform, i.e, $\phi(\mathbf{q}) = 1, \ \forall \mathbf{q} \in Q$. A backstepping controller will be designed to control agents with second order dynamics of the form

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{v}_i \\ \dot{\mathbf{v}}_i &= \bar{\mathbf{u}}_i\end{aligned} \tag{158}$$

where $\mathbf{x}_i$ is the position, $\mathbf{v}_i$ is the velocity, and $\bar{\mathbf{u}}_i$ is the control input of the i^{th} agent.

Theorem 5. . *Assume uniform density $\phi(\boldsymbol{q}) = 1,\ \forall \boldsymbol{q} \in Q$ and consider agents with second order dynamics (158). Then a control law which guarantees the convergence of the position of the agents to a centroidal Voronoi configuration is*

$$\bar{\boldsymbol{u}}_i = -\sqrt{\frac{s_i}{r_i}}\frac{d}{dt}\int_{V_i} \boldsymbol{v}_i \phi(\boldsymbol{q})\,\mathrm{d}\boldsymbol{q} - 2\int_{V_i} (\boldsymbol{x}_i - \boldsymbol{q})\phi(\boldsymbol{q})\,\mathrm{d}\boldsymbol{q} \tag{159}$$

where $s_i > 0$, $r_i > 0$.

Proof. According to the result in (156), let $\tilde{\mathbf{v}}_i$ be the desired velocity

$$\tilde{\mathbf{v}}_i = -\sqrt{\frac{s_i}{r_i}}\int_{V_i} (\mathbf{x}_i - \mathbf{q})\phi(\mathbf{q})\,\mathrm{d}\mathbf{q} \tag{160}$$

Let $\mathbf{z}_i$ be the tracking error between the velocity and the desired velocity

$$\mathbf{z}_i = \mathbf{v}_i - \tilde{\mathbf{v}}_i \tag{161}$$

We can express the dynamics in (158) in terms of $\mathbf{z}_i$ and $\tilde{\mathbf{v}}_i$ using (161) as

$$\dot{\mathbf{x}}_i = \mathbf{z}_i + \tilde{\mathbf{v}}_i \tag{162}$$

The dynamics of the tracking error are

$$\dot{\mathbf{z}}_i = \bar{\mathbf{u}}_i + \sqrt{\frac{s_i}{r_i}}\frac{d}{dt}\int_{V_i} (\mathbf{x}_i - \mathbf{q})\phi(\mathbf{q})\,\mathrm{d}\mathbf{q} \tag{163}$$

Let us consider the value function (157) from [34] as an initial candidate Lyapunov function

$$V_1(\mathbf{x}) = \sum_{i=1}^{n} \int_{V_i} ||\mathbf{x}_i - \mathbf{q}||^2 \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} \tag{164}$$

We can augment the candidate Lyapunov function in equation (164) by adding a term on the tracking error $V_2 = \sum_{i=1}^{n} \frac{1}{2} \mathbf{z}_i^T \mathbf{z}_i$, yielding

$$V(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} \int_{V_i} ||\mathbf{x}_i - \mathbf{q}||^2 \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} + \frac{1}{2} \mathbf{z}_i^T \mathbf{z}_i \tag{165}$$

Taking the time derivative of (165) yields

$$\dot{V}(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} \dot{\mathbf{x}}_i^T \frac{\partial}{\partial \mathbf{x}_i} \int_{V_i} ||\mathbf{x}_i - \mathbf{q}||^2 \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} + \mathbf{z}_i^T \dot{\mathbf{z}}_i \tag{166}$$

which is obtained using chain rule. Using the result of the partial derivative from (152), substituting $\dot{\mathbf{x}}_i = \mathbf{v}_i = \mathbf{z}_i + \tilde{\mathbf{v}}_i$ and the dynamics of the tracking error from (163), (166) can be expanded as follows

$$\begin{aligned} \dot{V}(\mathbf{x}, \mathbf{z}) = & -2 \sum_{i=1}^{n} \sqrt{\frac{s_i}{r_i}} || \int_{V_i} (\mathbf{x}_i - \mathbf{q}) \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} ||^2 \\ & + 2\mathbf{z}_i^T \int_{V_i} (\mathbf{x}_i - \mathbf{q}) \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} \\ & + \mathbf{z}_i^T \left(\bar{\mathbf{u}}_i + \sqrt{\frac{s_i}{r_i}} \frac{d}{dt} \int_{V_i} (\mathbf{x}_i - \mathbf{q}) \phi(\mathbf{q}) \, \mathrm{d}\mathbf{q} \right) \end{aligned} \tag{167}$$

For $V(\mathbf{x}, \mathbf{z})$ to be a Lyapunov function, the following conditions must be satisfied [61]

$$\begin{aligned} &1.\, V(\mathbf{X}) > 0, \, \forall \mathbf{X} \neq 0 \\ &2.\, V(\mathbf{X}) = 0, \, \mathbf{X} = 0 \\ &3.\, \dot{V}(\mathbf{X}) < 0, \, \forall \mathbf{X} \neq 0 \\ &4.\, \dot{V}(\mathbf{X}) = 0, \, \mathbf{X} = 0 \end{aligned} \tag{168}$$

where $\mathbf{X} = [\mathbf{x} \;\; \mathbf{z}]^T$. Note that the function (165) is always non-negative and it is only zero when $\mathbf{X}$ is such that the agents are located at the center of mass of their respective Voronoi cells (i.e, $\mathbf{x}_i = \mathbf{CM}_{V_i}$) and the velocity tracking error is zero. Let us define $\mathcal{X} = \{\mathbf{x} \;:\; \mathbf{x}_i = \mathbf{CM}_{V_i}, \; i = 1, \ldots, n\}$

and $\mathcal{S} = \mathcal{X} \times \{0\}$. For V to be a control Lyapunov function one must select $\bar{\mathbf{u}}_i$ such that V is positive and its derivative is negative unless $\mathbf{X} \in \mathcal{S}$. The claim that V is positive unless $X \in S$ is trivial to prove and does not depend on $\bar{\mathbf{u}}_i$. Therefore, $\bar{\mathbf{u}}_i$ will now be selected to ensure that $\dot{V}$ is negative unless $\mathbf{X} \in \mathcal{S}$. Let the control input be

$$\bar{\mathbf{u}}_i = -\sqrt{\frac{s_i}{r_i}} \frac{d}{dt} \int_{V_i} (\mathbf{x}_i - \mathbf{q})\phi(\mathbf{q})\, \mathrm{d}\mathbf{q} - 2 \int_{V_i} (\mathbf{x}_i - \mathbf{q})\phi(\mathbf{q})\, \mathrm{d}\mathbf{q} \tag{169}$$

Substituting (169) into (167) yields

$$\dot{V}_{aug}(\mathbf{x}, \mathbf{z}) = -2 \sum_{i=1}^{n} \sqrt{\frac{s_i}{r_i}} || \int_{V_i} (\mathbf{x}_i - \mathbf{q})\phi(\mathbf{q})\, \mathrm{d}\mathbf{q}||^2 \tag{170}$$

which is always non-positive since $r_i > 0$ and $s_i > 0$. The set where $\dot{V} = 0$ is $\mathcal{M} = \{(\mathbf{x}, \mathbf{z}) \ : \ \mathbf{x} \in \mathcal{X}\}$. For trajectories on this set we can see from the dynamics and the definition of the control input (159) that both $\mathbf{v}_i = 0$ and $\tilde{\mathbf{v}}_i = 0$, which makes $z_i = 0$. Therefore, using a LaSalle's argument [47] one can show that the largest invariant set contained in $\mathcal{M}$ is the set of positions of the agents corresponding to the centroids of the respective Voronoi cell and $\mathbf{x}_i = \tilde{\mathbf{v}}_i,\ i = 1, \ldots, n$, which finishes the proof. □

It is important to note that one must be consider the robustness of the backstepping technique when applying it to real systems. Due to the cancellation in terms, any uncertainty in system parameters can potentially lead to instability, and therefor it is prudent to first check within simulation. To implement the control input in (169), we must evaluate the time derivative in the expression. The term can be re-arranged as illustrated in (153)

$$\begin{aligned} \frac{d}{dt} \int_{V_i} (\mathbf{x}_i - \mathbf{q})\phi(\mathbf{q})\, \mathrm{d}\mathbf{q} &= \frac{d}{dt}\Big(M_{V_i}(\mathbf{x}_i - \mathbf{CM}_{V_i}) \Big) \\ &= \dot{M}_{V_i}(\mathbf{x}_i - \mathbf{CM}_{V_i}) + M_{V_i}(\mathbf{v}_i - \dot{\mathbf{CM}}_{V_i}) \end{aligned} \tag{171}$$

The time derivatives for the mass and the center of mass of the Voronoi cell can be expressed as follows

$$\begin{aligned}\dot{M}_{V_i} &= \frac{\partial M_{V_i}}{\partial t} + \frac{\partial M_{V_i}}{\partial \mathbf{x}_i}\dot{\mathbf{x}}_i + \sum_{j \in \mathcal{N}(i)} \frac{\partial M_{V_i}}{\partial \mathbf{x}_j}\dot{\mathbf{x}}_j \\ \mathbf{C}\dot{\mathbf{M}}_{V_i} &= \frac{\partial \mathbf{CM}_{V_i}}{\partial t} + \frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_i}\dot{\mathbf{x}}_i + \sum_{j \in \mathcal{N}(i)} \frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_j}\dot{x}_j\end{aligned} \tag{172}$$

The mass and center of mass will not be explicit functions of time so we can simplify (172) as follows:

$$\begin{aligned}\dot{M}_{V_i} &= \frac{\partial M_{V_i}}{\partial \mathbf{x}_i}\dot{\mathbf{x}}_i + \sum_{j \in \mathcal{N}(i)} \frac{\partial M_{V_i}}{\partial \mathbf{x}_j}\dot{\mathbf{x}}_j \\ \mathbf{C}\dot{\mathbf{M}}_{V_i} &= \frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_i}\dot{\mathbf{x}}_i + \sum_{j \in \mathcal{N}(i)} \frac{\partial \mathbf{CM}_{V_i}}{\partial \mathbf{x}_j}\dot{\mathbf{x}}_j\end{aligned} \tag{173}$$

The assumption of constant uniform density ($\phi(\mathbf{q}) = 1,\ \forall \mathbf{q} \in Q$), $A_{V_i} = M_{V_i}$, allows the substitution of (118), (107), (142), and (149) into (173)

$$\begin{aligned}\dot{M}_{V_i} &= \sum_{j \in \mathcal{N}(i)} l_{ij}\left(\frac{c_{ij}}{R_{ij}}\mathbf{b}_{ij} + \frac{1}{2}\mathbf{n}_{ij}\right)\dot{\mathbf{x}}_i + \sum_{j \in \mathcal{N}(i)} -l_{ij}\left(\frac{c_{ij}}{R_{ij}}\mathbf{b}_{ij} - \frac{1}{2}\mathbf{n}_{ij}\right)\dot{\mathbf{x}}_j \\ \mathbf{C}\dot{\mathbf{M}}_{V_i} &= \sum_{j \in \mathcal{N}(i)} \left(\frac{l_{ij}}{M_{V_i}}\mathbf{CM}_{V_i}\begin{bmatrix}\frac{c_{ij}}{R_{ij}} & \frac{1}{2}\end{bmatrix} + \frac{l_{ij}}{M_{V_i}}\begin{bmatrix}\frac{1}{R_{ij}}\left(\frac{l_{ij}^2}{12} + c_{ij}^2\right) & \frac{c_{ij}}{2} \\ 0 & 0\end{bmatrix}\right)\dot{\mathbf{x}}_i \\ &\quad + \sum_{j \in \mathcal{N}(i)} \left(\frac{l_{ij}}{M_{V_i}}\mathbf{CM}_{V_i}\begin{bmatrix}\frac{c_{ij}}{R_{ij}} & -\frac{1}{2}\end{bmatrix} + \frac{l_{ij}}{M_{V_i}}\begin{bmatrix}\frac{1}{R_{ij}}\left(-\frac{l_{ij}^2}{12} - c_{ij}^2\right) & \frac{c_{ij}}{2} \\ 0 & 0\end{bmatrix}\right)\dot{\mathbf{x}}_j\end{aligned} \tag{174}$$

Substituting (171) into (169) yields the following control law

$$\bar{\mathbf{u}}_i = -\sqrt{\frac{s_i}{r_i}}\left(\dot{M}_{V_i}(\mathbf{x}_i - \mathbf{CM}_{V_i}) + M_{V_i}(v_i - \mathbf{C}\dot{\mathbf{M}}_{V_i})\right) - 2M_{V_i}(\mathbf{x}_i - \mathbf{CM}_{V_i}) \tag{175}$$

where $\dot{M}_{V_i}$ and $\mathbf{C}\dot{\mathbf{M}}_{V_i}$ are defined in (174).

4.5 Simulation Results

This section shows the results of a Matlab simulation of the multi-agent coverage problem using the derived result for agents with second order dynamics, and provides a comparison of the results of the previous work described in section 4.4.2. Consider five agents, with zero initial velocity, placed randomly in a convex area to provide coverage. The agents will use the following control input from equation (175):

$$\bar{\mathbf{u}}_i = -\sqrt{\frac{s_i}{r_i}}\left(\dot{M}_{V_i}(\mathbf{x}_i - \mathbf{CM}_{V_i}) + M_{V_i}(v_i - \dot{\mathbf{CM}}_{V_i})\right) - 2M_{V_i}(\mathbf{x}_i - \mathbf{CM}_{V_i})$$

Figures 4.5 and 4.6 illustrate the paths each agent has taken, ending at the center of mass of their Voronoi cells, for agents with second order and first order dynamics (as covered in section 4.4.2) respectively. Figures 4.7 and 4.8 are plots of the speed of each agent, for agents of second order and first order dynamics respectively. In both control schemes the Voronoi diagrams converge to the

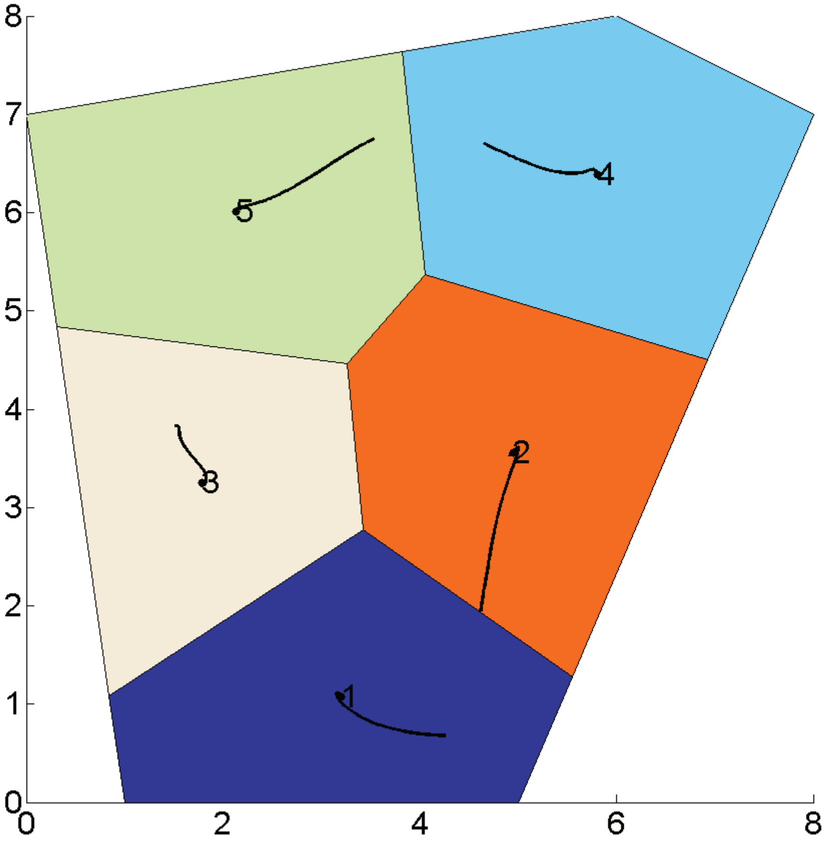


Figure 4.5: The Voronoi diagram at the final time of the simulation of agents with second order dynamics. The trajectories of the three agents are denoted by the black lines, ending at the black dot.

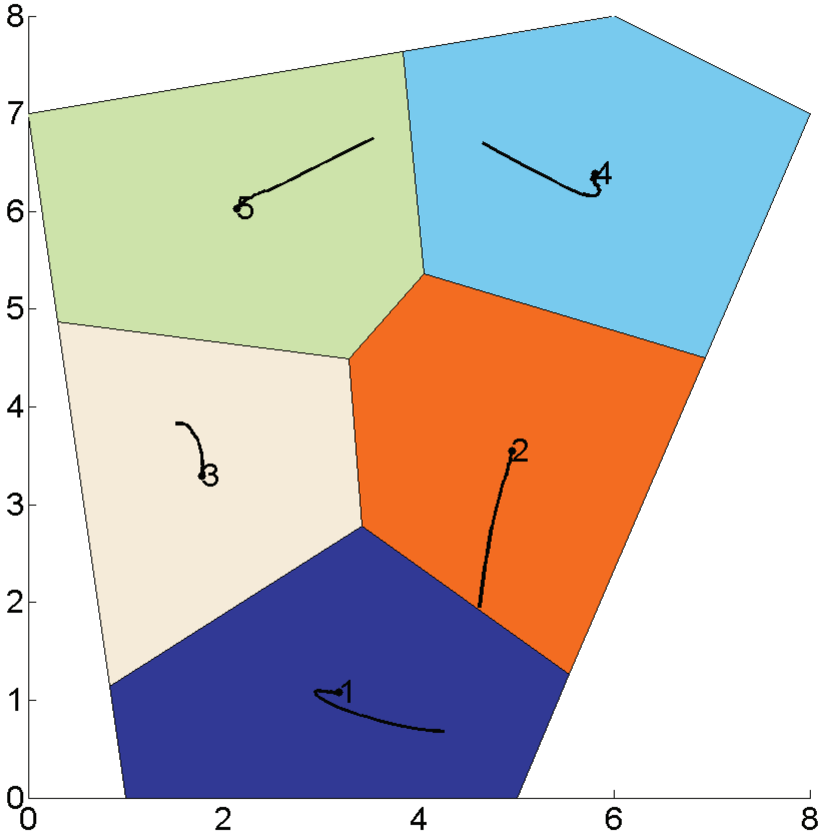


Figure 4.6: The Voronoi diagram at the final time of the simulation of agents with first order dynamics. The trajectories of the three agents are denoted by the black lines, ending at the black dot.

same CVC, as is expected for agents beginning in the same position between the two simulations. The speeds of the agent's (see figures 4.7 and 4.8) are very different between the first and second order control schemes. Using the first order controller, the initial speeds are very high due to the error between each agent and its Voronoi cell's center of mass. As expected the agents with second order dynamics converge at a much slower rate than the first order agents. The difference in trajectories of the agents between the two simulations can be attributed to the terms in the control law based on the relative velocity between an agent and its cell's center of mass, as well as the time rate of change of the mass of its cell. It is of particular interest to note the term $M_{V_i}(\mathbf{v}_i - \dot{\mathbf{CM}}_{V_i})$ which is a tracking error between the velocities of the agent and center of mass, weighted by the mass of

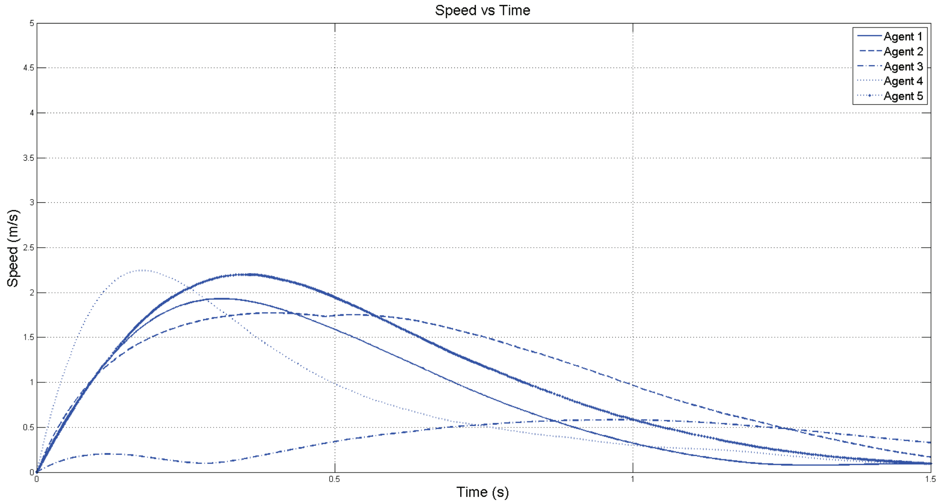


Figure 4.7: The speed of each agent with second order dynamics.

the Voronoi cell.

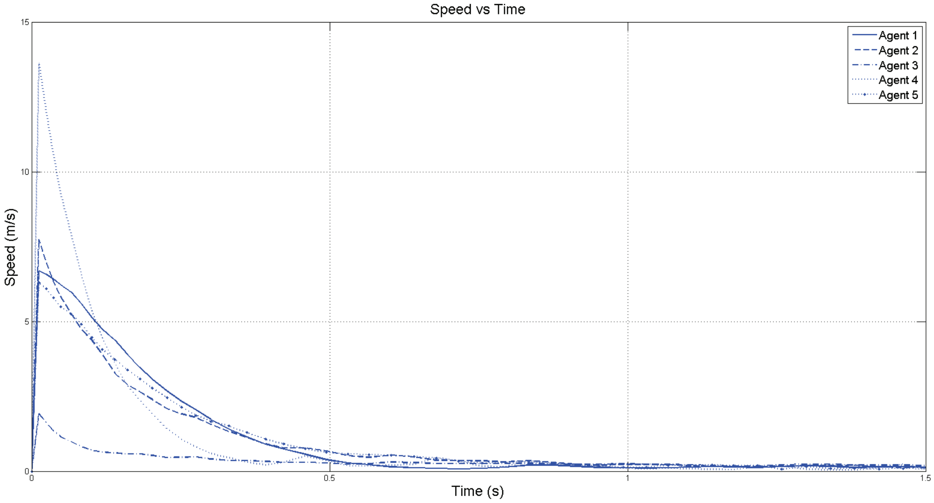


Figure 4.8: The speed of each agent with first order dynamics.

Chapter 5

Conclusions

This thesis has presented solutions to UAV flight system problems in the form of an optimal control problem for a single quadrotor UAV, a prototype software FMS as a research platform using industry standard communication protocols, and a multi-agent coverage problem using agents with second order dynamics.

The quadrotor optimal control problem considers a cost index, as found in commercial flight management systems, to provide a trade-off between time and a measure of control effort. The use of a cost index allows operators of UAVs to minimize the cost of their flight operations much like many commercial airlines currently do. The proposed formulation allows a range of cost indices to be chosen according to the maximum velocity attainable by the quadrotor. The solution to the proposed problem yields an analytic state-feedback control law.

A software FMS implementation has been presented with experimental results using an industry flight simulator. The SFMS provides some standard functionality found in commercial FMSs such as flight plan creation, lateral navigation, and a performance page (the usage of a cost index). The SFMS uses aerospace standard communication protocols ARINC 429 and ARINC 834 to interface with an aircraft/simulator. The SFMS was verified by interfacing the SFMS with an industry flight simulator to perform a simulated flight.

A multi-agent coverage problem using agents with second order dynamics has been considered. Backstepping was used to extend the result of an optimal control formulation of a previously solved coverage problem to agents with second order dynamics. To compute the resultant control law, the change in area and change in position of the center of mass of Voronoi cells due to moving agents is required. A geometric solution has been presented for both the change in area and change in position of the center of mass of Voronoi cells. Furthermore, the change in position of the center of mass has also been solved using Reynolds Transport Theorem. Simulated results have been presented, showing the comparison between the first order agents and second order agents.

The work in this thesis has outlined some important upcoming topics for UAVs, from the optimization of a quadrotor's trajectory for package delivery to the usage of multiple UAVs in multi-agent systems for search and rescue or surveillance. We have considered an SFMS implementation which can be used aboard a UAV to provide similar functionality as found for manned aircraft. Using aerospace standard communication protocols for the SFMS allows for easier future integration of UAVs and aircraft in shared airspace.

With UAVs set to begin operating in larger numbers in the coming years, we have found that more attention must be focused on the efficiency and cost of operating UAVs in both single and multi-agent systems. One can look to current manned aircraft for inspiration, such as the use of a cost index to minimize operating costs. The widespread usage of UAVs in commercial applications (i.e. package delivery and unmanned passenger transport) will encourage researchers to address problems involving both manned and unmanned aerial vehicles operating within not only a shared physical airspace, but also within the same communication network. One can imagine a fully autonomous system in the future in which all aerial vehicles are part of a single network that will lead to the formulation of new cost problems.

Bibliography

[1] J. Villarroel, "An optimal control framework for flight management systems," Master's thesis, Concordia University, 1455 De Maisonneuve Blvd. W. Montreal, QC, Canada, 2 2015.

[2] R. S. Shevell, *Fundamentals of Flight*. Prentice Hall, 2 ed., 1988.

[3] V. Dobrokhodov, *Handbook of Unmanned Aerial Vehicles*, ch. 14, pp. 243–277. Springer Netherlands, 2015.

[4] "Arinc 429 specification tutorial," tech. rep., AIM GmbH, 2010.

[5] *Aircraft Data Interface Function: ARINC Specification 834-1*, 2009.

[6] World Health Organization, "Climate change and human health - risks and responses. Summary."

[7] Air Transport Action Group, "Aviation benefits beyond borders."

[8] Internation Civil Aviation Organization, "Aircraft engine emissions."

[9] BI Intelligence, "The drones report: Market forecasts, regulatory barriers, top vendors, and leading commercial applications," June 2016. [Online; posted June-2016].

[10] E. C. Suicmez and A. T. Kutay, "Optimal path tracking control of a quadrotor uav," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pp. 115–125, May 2014.

[11] W. Hadjadj-Aoul, A. Mokhtari, and A. Benallegue, “Optimal control using backstepping technique of a quadrotor helicopter,” in *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, pp. 1542–1547, Dec 2014.

[12] R. Zawiski and M. Bachuta, “Model development and optimal control of quadrotor aerial robot,” in *Methods and Models in Automation and Robotics (MMAR), 2012 17th International Conference on*, pp. 475–480, Aug 2012.

[13] M. Geisert and N. Mansard, “Trajectory generation for quadrotor based systems using numerical optimal control,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2958–2964, May 2016.

[14] F. Morbidi, R. Cano, and D. Lara, “Minimum-energy path generation for a quadrotor uav,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1492–1498, May 2016.

[15] A. de Souza Cndido, R. K. H. Galvo, and T. Yoneyama, “Control and energy management for quadrotor,” in *Control (CONTROL), 2014 UKACC International Conference on*, pp. 343–348, July 2014.

[16] R. Ritz, M. Hehn, S. Lupashin, and R. D’Andrea, “Quadrocopter performance benchmarking using optimal control,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5179–5186, Sept 2011.

[17] K. Eliker, H. Bouadi, and M. Haddad, “Flight planning and guidance features for an uav flight management computer,” in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–6, Sept 2016.

[18] J. Jamieson and J. Biggs, “Path planning using concatenated analytically-defined trajectories for quadrotor uavs,” *Aerospace*, vol. 2, no. 2, pp. 155–170, 2015.

[19] M. Radmanesh, M. Kumar, A. Nemati, and M. Sarim, “Dynamic optimal uav trajectory planning in the national airspace system via mixed integer linear programming,” *Proceedings of*

the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, vol. 230, no. 9, pp. 1668–1682, 2016.

[20] Y. Bouktir, M. Haddad, and T. Chettibi, “Trajectory planning for a quadrotor helicopter,” in *2008 16th Mediterranean Conference on Control and Automation*, pp. 1258–1263, June 2008.

[21] M. W. Mueller, M. Hehn, and R. D’Andrea, “A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasibility verification,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3480–3486, Nov 2013.

[22] K. A. Klein, H. Eckberg, and R. H. Dean, “An analysis of low-cost simulated flight management systems for aviation research,” in *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*, pp. 1.C.3–1–1.C.3–9, Oct 2009.

[23] J. W. Adams and C. G. Merrett, “Combined flight management system and flight data recorder for general aviation using tablet computers,” in *AIAA Infotech @ Aerospace, AIAA SciTech*, 2016.

[24] T. J. Koo, B. Sinopoli, A. Sangiovanni-Vincentelli, and S. Sastry, “A formal approach to reactive system design: unmanned aerial vehicle flight management system design example,” in *Computer Aided Control System Design, 1999. Proceedings of the 1999 IEEE International Symposium on*, pp. 522–527, 1999.

[25] A. F. Tarhan, E. Koyuncu, M. Hasanzade, U. Ozdemir, and G. Inalhan, “Formal intent based flight management system design for unmanned aerial vehicles,” in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pp. 984–992, May 2014.

[26] J. U. Inc., “Sparrow flight management system,” 2017.

[27] C. Torens, F.-M. Adolf, and L. Goormann, “Certification and software verification considerations for autonomous unmanned aircraft,” *Journal of Aerospace Information Systems*, vol. 11, pp. 649–664, 2014.

[28] J. Villarroel and L. Rodrigues, “Optimal control framework for cruise economy mode of flight management systems,” *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 5, pp. 1022–1033, 2016.

[29] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. New York, NY, USA: John Wiley & Sons, Inc., 1992.

[30] P. Español and M. Serrano, “Voronoi fluid particles & tessellation fluid dynamics.” Unpublished, 2009.

[31] Q. Du, V. Faber, and M. Gunzburger, “Centroidal voronoi tessellations: Applications and algorithms,” *SIAM REV*, vol. 41, no. 4, pp. 637–676, 1999.

[32] J. Cortes, S. Martinez, and F. Bullo, “Spatially-distributed coverage optimization and control with limited-range interactions,” *ArXiv Mathematics e-prints*, Jan. 2004.

[33] Y. Diaz-Mercado, S. G. Lee, and M. Egerstedt, “Distributed dynamic density coverage for human-swarm interactions,” in *2015 American Control Conference (ACC)*, pp. 353–358, July 2015.

[34] M. Moarref and L. Rodrigues, “An optimal control approach to decentralized energy-efficient coverage problems,” {*IFAC*} *Proceedings Volumes*, vol. 47, no. 3, pp. 6038 – 6043, 2014. 19th {IFAC} World Congress.

[35] M. Pavone, E. Frazzoli, and F. Bullo, “Distributed policies for equitable partitioning: Theory and applications,” in *2008 47th IEEE Conference on Decision and Control*, pp. 4191–4197, Dec 2008.

[36] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, “Distributed algorithms for environment partitioning in mobile robotic networks,” *IEEE Transactions on Automatic Control*, vol. 56, pp. 1834–1848, Aug 2011.

[37] K. Laventall and J. Cortes, “Coverage control by robotic networks with limited-range anisotropic sensory,” in *2008 American Control Conference*, pp. 2666–2671, June 2008.

[38] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, "Dynamic modelling and configuration stabilization for an x4-flyer," in *2002 International Federation of Automatic Control (IFAC)*, vol. 35, pp. 217–222, 2002.

[39] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, pp. 20–32, Sept 2012.

[40] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, pp. 72–77 vol.1, 2002.

[41] I. C. Dikmen, A. Arisoy, and H. Temeltas, "Attitude control of a quadrotor," in *Recent Advances in Space Technologies, 2009. RAST '09. 4th International Conference on*, pp. 722–727, June 2009.

[42] D. E. Kirk, *Optimal Control Theory: An Introduction*. Englewood Cliffs, NJ: Prentice-Hall Inc., 1970.

[43] A. Bryson and Y. Ho, *Applied Optimal Control*. New York, NY: Talyor & Francis Group, 1975.

[44] R. Walter, *The Avionics Handbook*, ch. 15. CRC Press, 2001.

[45] D. G. Hull, *Fundamentals of Airplane Flight Mechanics*. Springer, 2007.

[46] A. Micaelli, C. Samson, and P. Icare, "Trajectory tracking for unicycle-type and two-steering-wheels mobile robots," 1993.

[47] H. K. Khalil, *Nonlinear Systems*, ch. 4. Upper Saddle Rivier, New Jersey: Prentice Hall, 3 ed., 2002.

[48] *Order 8260.54A*, 2007.

[49] T. R. Yechout et al., *Introduction to Aircraft Flight Mechanics*. Reston, Virginia: AIAA, 2003.

[50] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*, ch. 2. McGraw-Hill, 2008.

[51] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, ch. 2. Artech House, 2008.

[52] M. Kayton, *Avionics Navigation Systems*, ch. 2. John Wiley & Sons, Inc., 2 ed., 1997.

[53] I. Todhunter, *Spherical Trigonometry*. Macmillan and Co., 5 ed., 1886.

[54] *ARINC 429: General Aviation Subset*, 2009.

[55] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ, USA: Princeton University Press, 2009.

[56] M. Gurtin, "Kinematics," in *An Introduction to Continuum Mechanics*, pp. 78 – 79, New York: Academic Press, 1 ed., 1981.

[57] B. Munson, D. Young, and T. Okiishi, *Fundamentals of Fluid Mechanics*. Hoboken: John Wiley, 6 ed., 2009.

[58] C. Truesdell and R. Toupin, *The Classical Field Theories*, pp. 226–858. Berlin, Heidelberg: Springer Berlin Heidelberg, 1960.

[59] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 243–255, April 2004.

[60] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, Mar 1982.

[61] H. K. Khalil, *Nonlinear systems*. Upper Saddle River, (N.J.): Prentice Hall, 2 ed., 1995.