Quadcopter Attitude Control Optimization and Multi-Agent Coordination

John McCormack
University of New Hampshire, Durham

Follow this and additional works at: https://scholars.unh.edu/thesis

Recommended Citation
McCormack, John, "Quadcopter Attitude Control Optimization and Multi-Agent Coordination" (2019). Master's Theses and Capstones. 1277.
https://scholars.unh.edu/thesis/1277

This Thesis is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Master's Theses and Capstones by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.
QUADCOPTER ATTITUDE CONTROL OPTIMIZATION AND MULTI-AGENT COORDINATION

BY

JOHN MCCORMACK

BS, Mechanical Engineering, University of New Hampshire, 2016

THESIS

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Master of Science
in
Mechanical Engineering

May 2019
This dissertation has been examined and approved in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering by:

**Committee Chair, May-Win Thein,**
Associate Professor of Mechanical Engineering

**Se Young Yoon,**
Assistant Professor of Electrical and Computer Engineering

**Michael Carter,**
Associate Professor of Electrical and Computer Engineering

on May 14th, 2019.

Original approval signatures are on file with the University of New Hampshire Graduate School.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>vi</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>x</td>
</tr>
</tbody>
</table>

CHAPTER

1. INTRODUCTION                                                                 | 1   |

1.1 Particle Swarm Optimization                                                  | 1   |
1.2 Robotic Swarm Control Methods                                               | 2   |
1.3 Organization                                                               | 3   |

2. QUADCOPTER MODELING AND CONTROL                                             | 4   |

2.1 Quadcopter Model                                                           | 4   |
2.2 Sliding Mode Control                                                        | 7   |

3. QUADCOPTER CONTROLLER GAIN OPTIMIZATION                                     | 9   |

3.1 Particle Swarm Optimization                                                 | 9   |
3.2 SMC Gain Tuning                                                            | 11  |
3.3 Simulation Results and Discussion                                           | 14  |

4. ARTIFICIAL POTENTIAL FIELD SWARM COORDINATION                                | 17  |

4.1 Leader-Follower Framework                                                   | 17  |
4.2 Formulation                                                                | 18  |
4.2.1 Decentralization                                                         | 18  |
4.2.2 Balancing Forces                                                         | 18  |
4.2.3 Smoothness                                                               | 19  |
4.3 Equilibrium Distance                                                       | 21  |
4.4 Stability Under First-Order Dynamics                                         | 28  |
4.4.1 Swarm Centroid Dynamics ................................................................. 28
4.4.2 Agent Dynamics .................................................................................. 32

4.5 Stability Under General Dynamics ......................................................... 34
4.6 Simulation Results and Discussion ......................................................... 39
  4.6.1 Quadcopter Swarm ........................................................................... 39
  4.6.2 Point-Mass Swarms ......................................................................... 44

5. EXPERIMENTAL DEVELOPMENT ......................................................... 49

6. CONCLUSION AND FUTURE WORK ...................................................... 52

BIBLIOGRAPHY .......................................................................................... 54

APPENDIX: PARTICLE SWARM OPTIMIZATION FIGURES ......................... 57
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>System Parameters</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>SMC Gain Search Bounds on $k_i$</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>SMC Gain Search Bounds on $c_i$</td>
<td>8</td>
</tr>
<tr>
<td>3.1</td>
<td>PSO Tuned Quadcopter SMC Gains with Coupled Responses</td>
<td>14</td>
</tr>
<tr>
<td>3.2</td>
<td>PSO Tuned Quadcopter SMC Gains with Decoupled Responses</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>PSO Tuned Quadcopter SMC Gains from Coupled Responses and MSE Cost</td>
<td>15</td>
</tr>
<tr>
<td>3.4</td>
<td>Optimized SMC Gains</td>
<td>16</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Quadrotor Model</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>Visualization of varying PSO gains</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Visualization of reflect boundary technique</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Cost convergence of 30 particle over 100 steps</td>
<td>16</td>
</tr>
<tr>
<td>4.1</td>
<td>Multi-agent swarm aggregation</td>
<td>19</td>
</tr>
<tr>
<td>4.2</td>
<td>Artificial Potential Field visualization in a single dimension</td>
<td>20</td>
</tr>
<tr>
<td>4.3</td>
<td>Real branches of the Lambert-W Function</td>
<td>22</td>
</tr>
<tr>
<td>4.4</td>
<td>Trajectories aggregating a 4-node formation</td>
<td>40</td>
</tr>
<tr>
<td>4.5</td>
<td>Trajectories tracking a 2-node formation</td>
<td>41</td>
</tr>
<tr>
<td>4.6</td>
<td>Quadcopter swarm trajectories before and after agent failure</td>
<td>42</td>
</tr>
<tr>
<td>4.7</td>
<td>Agent coordinates aggregating a 2 node formation with single-agent failure</td>
<td>43</td>
</tr>
<tr>
<td>4.8</td>
<td>Lawnmower path achieved through waypoint navigation</td>
<td>43</td>
</tr>
<tr>
<td>4.9</td>
<td>Large swarms with formation and point mass dynamics</td>
<td>46</td>
</tr>
<tr>
<td>4.10</td>
<td>8-agent swarms with 4-node formation and point mass dynamics</td>
<td>46</td>
</tr>
<tr>
<td>4.11</td>
<td>Large swarms with point mass dynamics and varying equilibrium distance</td>
<td>47</td>
</tr>
<tr>
<td>4.12</td>
<td>Large swarms with formation and point mass dynamics with limited</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>communication</td>
<td></td>
</tr>
<tr>
<td>4.13</td>
<td>Inline formation with and without assigning agents to specific nodes</td>
<td>48</td>
</tr>
<tr>
<td>5.1</td>
<td>Attitude response using PSO tuned SMC controller</td>
<td>50</td>
</tr>
</tbody>
</table>
A.1 Cost of each particle’s controller at each step using equation (3.3) as the cost function and coupled responses ($N = 10, k_f = 100$) ....................... 58

A.2 Controller gains for each particle at each step using equation (3.3) as the cost function and coupled responses ($N = 10, k_f = 100$) ....................... 58

A.3 Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and coupled responses ($N = 10, k_f = 100$) .............. 59

A.4 Cost of each particle’s controller at each step using equation (3.3) as the cost function and coupled responses ($N = 100, k_f = 10$) ....................... 59

A.5 Controller gains for each particle at each step using equation (3.3) as the cost function and coupled responses ($N = 100, k_f = 10$) ....................... 60

A.6 Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and coupled responses ($N = 100, k_f = 10$) .............. 60

A.7 Cost of each particle’s controller at each step using equation (3.3) as the cost function and coupled responses ($N = 30, k_f = 100$) ....................... 61

A.8 Controller gains for each particle at each step using equation (3.3) as the cost function and coupled responses ($N = 30, k_f = 100$) ....................... 61

A.9 Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and coupled responses ($N = 30, k_f = 100$) .............. 61

A.10 Cost of each particle’s controller at each step using equation (3.3) as the cost function and decoupled responses ($N = 10, k_f = 100$) ....................... 62

A.11 Controller gains for each particle at each step using equation (3.3) as the cost function and decoupled responses ($N = 10, k_f = 100$) ....................... 62

A.12 Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and decoupled responses ($N = 10, k_f = 100$) .............. 63

A.13 Cost of each particle’s controller at each step using equation (3.3) as the cost function and decoupled responses ($N = 100, k_f = 10$) ....................... 63

A.14 Controller gains for each particle at each step using equation (3.3) as the cost function and decoupled responses ($N = 100, k_f = 10$) ....................... 64

A.15 Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and decoupled responses ($N = 100, k_f = 10$) .............. 65
A.16 Cost of each particle’s controller at each step using equation (3.3) as the cost function and decoupled responses ($N = 30, k_f = 100$) ......................... 65

A.17 Controller gains for each particle at each step using equation (3.3) as the cost function and decoupled responses ($N = 30, k_f = 100$) ......................... 66

A.18 Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and decoupled responses ($N = 30, k_f = 100$) ......................... 66

A.19 Cost of each particle’s controller at each step using equation (3.5) as the cost function and coupled responses ($N = 10, k_f = 100$) ......................... 67

A.20 Controller gains for each particle at each step using equation (3.5) as the cost function and coupled responses ($N = 10, k_f = 100$) ......................... 67

A.21 Quadcopter responses using the final optimized controller found using equation (3.5) as the cost function and coupled responses ($N = 10, k_f = 100$) ......................... 68

A.22 Cost of each particle’s controller at each step using equation (3.5) as the cost function and coupled responses ($N = 100, k_f = 10$) ......................... 68

A.23 Controller gains for each particle at each step using equation (3.5) as the cost function and coupled responses ($N = 100, k_f = 10$) ......................... 68

A.24 Quadcopter responses using the final optimized controller found using equation (3.5) as the cost function and coupled responses ($N = 100, k_f = 10$) ......................... 69

A.25 Cost of each particle’s controller at each step using equation (3.5) as the cost function and coupled responses ($N = 20, k_f = 100$) ......................... 70

A.26 Controller gains for each particle at each step using equation (3.5) as the cost function and coupled responses ($N = 30, k_f = 100$) ......................... 70

A.27 Quadcopter responses using the final optimized controller found using equation (3.5) as the cost function and coupled responses ($N = 30, k_f = 100$) ......................... 71
ABSTRACT

Quadcopter Attitude Control Optimization and Multi-Agent Coordination

by

John McCormack

University of New Hampshire, May, 2019

This thesis presents a method of automated control gain tuning for a Quadcopter Unmanned Aerial Vehicle and proposes a method of coordination multiple autonomous robotic agents capable for formation aggregation. Sliding Mode Control for Quadcopter altitude and attitude stabilization is presented and tuned using Particle Swarm Optimization. Different configurations for the optimization process are compared to determine an effective and time-efficient setup to complete the control gain tuning. The multi-agent coordination scheme expands upon an existing adjustable swarm framework based on an Artificial Potential Field Sliding Mode Controller. The original leader-follower scheme is modified with the goal of producing a leaderless swarm where agents move towards specific locations to aggregate a desired formation. Analysis of the swarm control scheme pays particular attention to maintaining proper distance between agents. Using Lyapunov methods following that of the original controller analysis, stability under first order and general higher order dynamics is analyzed. Numerical simulations of the swarm controller using agents with nonlinear Quadcopter or second order point mass dynamics are presented to illustrate the capabilities of this algorithm. The automatically tuned Quadcopter controller is used in simulations when applicable. The development of an experimental test platform is discussed with the intention of validating the simulation results on physical Quadcopters.
CHAPTER 1
INTRODUCTION

Applications of groups of autonomous robots, such as unmanned aerial vehicles (UAV), have grown rapidly in numerous fields such as agricultural monitoring, land surveying, and search and rescue operations [1, 2, 3]. Quadcopter UAVs in particular have become increasingly popular due to their high maneuverability and versatility in a wide range of tasks. Therefore, the UNH Advanced Controls Laboratory began the development of a custom Quadcopter platform to serve as a testbed for potential control techniques and experimental applications.

During development of this platform, it was found that tuning control gains was a time-consuming and non-trivial process due to the coupled and nonlinear nature of quadcopter dynamics. As such, an automated tuning process was desired. Additionally, a method of controlling the positions of multiple robotic agents was needed with the potential for formation aggregation. This thesis presents a potential solution to each of these problems. Firstly, Particle Swarm Optimization (PSO) is considered for automatic tuning of the Quadcopter control gains. Secondly, an Artificial Potential Field (APF) Sliding Mode Control (SMC) is considered for decentralized multi-agent coordination and formation aggregation.

1.1 Particle Swarm Optimization

PSO, first introduced in [4, 5] and developed further in [6, 7], was inspired by the movements of flocks of birds and schools of fish and has been found to be highly effective in solving complicated optimization problems [8, 9]. Genetic algorithms and evolutionary programming are well studied methods often used for comparison with PSO which has some notable advantages over those and other optimization techniques. First and foremost, PSO does not require any \textit{a priori} information about the search space or variables. It is, therefore, capable of executing a multi-variable optimiza-
tion of highly complicated systems with variables that may not be directly or evidently related in a hyper-spatial search region. PSO works to minimize a cost function by comparing multiple samples of the optimization variables in successive steps. The values of the optimization variables for the next step are then weighted to tend toward a set which gives the lowest cost. This is performed with any constant number of samples (or particles) for any number of steps until an end condition is met. Additionally, PSO is relatively simple when compared with other optimization methods and requires very little processing time and power.

Particle swarm optimization has proven to be an effective and efficient method of gain tuning for a number of systems and controller types [10, 11]; namely Fuzzy SMC for chaotic systems [12] and various PID controllers for quadcopters. Position PID gains were tuned in [13] while both position and attitude PID controllers were tuned simultaneously in [14]. A Fuzzy PID quadcopter attitude controller was tuned in [15]. In [16], a PSO-tuned PID position controller for formation control of multiple quadcopters was found to perform better than LQR control for the same system. Finally, a modified version of PSO was used in [17] and shown to simultaneously tune PID gains for quadcopter attitude and position.

This thesis proposes a novel cost function for PSO tuning of quadcopter altitude and attitude control gain. This cost function is composed of several parameters extracted from both regulation and tracking simulations which utilize the nonlinear and coupled dynamics of a quadcopter. The performance of resulting controllers from various PSO configurations are compared to determine the best PSO configuration while also considering the time required to complete the optimization.

1.2 Robotic Swarm Control Methods

Within these groups or swarms, agents must often work collaboratively to complete a task. Methods of coordination of agents is often inspired by studies of groups in nature such as flocks of birds or schools of fish. Flocking behavior in particular was considered in [18] where follower agents flock around, a leader following a set trajectory and [19] which incorporated a virtual leader to influence agents behavior.
A weakness of the flocking method of swarm coordination that is there exists very little control over agents’ relative positions. A predefined structure or formation may be needed. Methods such as virtual structures in [20] and finite-time consensus [21] are capable of aggregating a formation from a swarm. However, this generally requires assigning agents to specific target locations \textit{a priori} and lacks flocking behavior.

The use of an artificial potential field (APF) in the sliding manifold in sliding mode control (SMC), proposed in [22] and expanded on in [23], shows particular promise in swarm coordination for leader-follower flocking behavior [24] and robot navigation and obstacle avoidance [25]. This method also may be applied in satellite constellation aggregation [26].

This thesis expands upon the leader-follower APF-SMC framework presented by Fabian et al. in [24] with the goal of a multi-agent distributed control scheme capable of aggregating a time-varying formation while additional agents in a larger swarm flock in the vicinity of the formation. The inclusion of flocking agents with the formation introduces redundancy in the swarm enabling the continuation of group missions regardless of agent failure.

1.3 Organization

In Chapter 2 quadcopter dynamics are reviewed and a Sliding Mode Control scheme is presented for attitude and altitude control. Chapter 3 details the Particle Swarm Optimization process and the cost function used. Optimization results are also presented and discussed. The APF-SMC multi-agent coordinator is derived in Chapter 4 followed by a stability analysis which follows the process presented in [24]. Swarm simulations using both point mass agents and quadcopter dynamics are shown and discussed. In Chapter 5 the development of the experimental platform and implementation of the PSO-tuned controller and APF-SMC multi agent coordinator is discussed. Chapter 6 suggests future work and provides conclusions of the present research.
CHAPTER 2
QUADCOPTER MODELING AND CONTROL

In this chapter the kinematic and dynamic model for a Quadrotor Unmanned Aerial Vehicle (UAV) is reviewed and Sliding Mode Control (SMC) laws presented in [27] are described. This model and attitude controller is used in simulation to test automatic controller gain tuning and multi-agent coordination in later chapters.

2.1 Quadcopter Model

Modeling assumptions follow that of [27] and are itemized as follows:

1. The center of mass is located at the origin of the body coordinate axes which lie along the principle axes of inertia.

2. Generated thrust and drag are proportional to the square of propeller speed.

3. The quadrotor platform is flying in a closed, laboratory setting.

4. Modeling uncertainties are considered negligible.

Figure 2.1 represents the quadrotor model used in this study and in previous work [27]. Here, the motor configurations are shown, with the red propeller indicating the forward direction of the quadcopter. \( \Omega_i \) represents the rotor speed for motor \( i = 1,2,3,4 \).
The rotational kinematics and translational dynamics are:

\[
\begin{align*}
\ddot{\phi} &= b_1 U_2 + a_1 \dot{\theta} \dot{\psi} - a_2 \dot{\theta} \Omega_r, \\
\ddot{\theta} &= b_2 U_3 + a_3 \dot{\phi} \dot{\psi} + a_4 \dot{\psi} \Omega_r, \\
\ddot{\psi} &= b_3 U_4 + a_5 \dot{\phi} \dot{\theta}, \\
\ddot{x} &= \frac{U_1}{m} (\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta), \\
\ddot{y} &= \frac{U_1}{m} (\cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi), \\
\ddot{z} &= \frac{U_1}{m} (\cos \phi \cos \theta) - g.
\end{align*}
\]  

(2.1)

where \(U_1, U_2, U_3,\) and \(U_4\) are the thrust forces relating to heave, pitch, roll, and yaw, respectively. The state variables \(\phi, \theta, \psi, x, y, z\) represent pitch, roll, yaw, x-position, y-position, and altitude, respectively, where \(l\) is the arm length of the quadrotor and \(J_r\) is the rotor inertia. The system parameters are obtained through physical modeling of a prototype experimental platform and literature search [28, 29]. The parameters assigned to the simulated system are shown in Table 2.1. With the state vector and control input defined as \(X = [\dot{\phi} \dot{\theta} \dot{\psi} x \dot{x} y \dot{y} z \dot{z}]^T\) and \(U = [U_1 U_2 U_3 U_4]\).
Table 2.1: System Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>0.98 kg</td>
</tr>
<tr>
<td>g</td>
<td>9.81 m/s</td>
</tr>
<tr>
<td>l</td>
<td>0.17 m</td>
</tr>
<tr>
<td>$I_{xx}$</td>
<td>0.01548 kgm²</td>
</tr>
<tr>
<td>$I_{yy}$</td>
<td>0.01565 kgm²</td>
</tr>
<tr>
<td>$I_{zz}$</td>
<td>0.03024 kgm²</td>
</tr>
<tr>
<td>$J_r$</td>
<td>6e-5 kgm²</td>
</tr>
<tr>
<td>$K_f$</td>
<td>3.13e-5 N-s²</td>
</tr>
<tr>
<td>$K_m$</td>
<td>3.13e-5 N-m-s²</td>
</tr>
</tbody>
</table>

$U_3 U_4^T$, the dynamics represented in (2.1) are written as:

$$\dot{X} = f(X) + \Phi(X)\bar{U}(t) \quad (2.2)$$

where $f(X) \in \mathbb{R}^{12 \times 1}$ and $\Phi(X) \in \mathbb{R}^{12 \times 4}$ is the input coefficient matrix. The system inputs are related to rotor speeds as:

$$
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix}
= 
\begin{bmatrix}
K_f & K_f & K_f & K_f \\
0 & -K_f & 0 & K_f \\
K_f & 0 & -K_f & 0 \\
K_m & -K_m & K_m & -K_m
\end{bmatrix}
\begin{bmatrix}
\Omega_1^2 \\
\Omega_2^2 \\
\Omega_3^2 \\
\Omega_4^2
\end{bmatrix}
\quad (2.3)
$$

where $K_f$ and $K_m$ are the aerodynamic force and moment constants respectively. The relative rotor speed $\Omega_r$ is the sum of rotor speeds:

$$\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (2.4)$$

The parameters $a_i$ and $b_i$ are the normalized and simplified inertia terms of the quadrotor, respectively. They are defined as:
\[
\begin{align*}
a_1 &= \frac{I_{yy} - I_{zz}}{I_{xx}} \\
a_2 &= \frac{J_r}{I_{xx}} \\
a_3 &= \frac{I_{zz} - I_{xx}}{I_{yy}} \\
a_4 &= \frac{J_r}{I_{yy}} \\
a_5 &= \frac{I_{xx} - I_{yy}}{I_{zz}} \\
b_1 &= \frac{l}{I_{xx}} \\
b_2 &= \frac{l}{I_{yy}} \\
b_3 &= \frac{1}{I_{zz}}
\end{align*}
\] (2.5)

### 2.2 Sliding Mode Control

The SMC attitude and altitude control laws from [27] are

\[
\begin{align*}
U_1 &= \frac{m}{\cos \phi \cos \theta} [k_z \text{sgn}(s_z) + c_z (\dot{z} - \dot{z}_d) + g + \ddot{z}_d] \\
U_2 &= \frac{1}{b_1} [k_\phi \text{sgn}(s_\phi) + c_\phi (\dot{\phi}_d - \dot{\phi}) + \ddot{\phi}_d + a_2 \ddot{\Omega}_r - a_1 \dot{\theta} \dot{\psi}] \\
U_3 &= \frac{1}{b_2} [k_\theta \text{sgn}(s_\theta) + c_\theta (\dot{\theta}_d - \dot{\theta}) + \ddot{\theta}_d - a_4 \dot{\phi} \Omega_r - a_3 \dot{\phi} \dot{\psi}] \\
U_4 &= \frac{1}{b_3} [k_\psi \text{sgn}(s_\psi) + c_\psi (\dot{\psi}_d - \dot{\psi}) + \ddot{\psi}_d - a_5 \dot{\phi} \dot{\theta}]
\end{align*}
\] (2.6)

The sliding surface is defined as \( s_i = c_i e_i + \dot{e}_i \), where tracking error is \( e_i = X_{d,i} - X_i \) with \( i = \phi, \theta, \psi \) and \( z \). The equivalent and discontinuous control gains are \( c_i \) and \( k_i \), respectively. These control gains are selected through Particle Swarm Optimization (detailed in a later chapter) given a finite range. The lower of the discontinuous gains \( k_i \) is determined from the maximum system uncertainty to satisfy Lyapunov stability while the upper bound is selected to prevent excessive control action [28, 30, 29, 31]. Ranges for the equivalent control gains \( c_i \) were selected using a similar process. The gain ranges can be see in Tables 2.2 and 2.3.
The quadcopter inputs (2.6) control the roll, pitch, yaw, and altitude only while the $x$ and $y$ horizontal positions are dependent on the attitude and heave $U_1$. A multi-agent coordination algorithm detailed in a later chapter is used to determine the virtual inputs $u_x$, $u_y$, and $u_z$ which are used to determine the desired roll $\theta_d$ and pitch $\phi_d$ such that

$$\theta_d = \tan^{-1}\left(\frac{u_x \cos \psi + u_y \sin \psi}{u_z + g}\right)$$

$$\phi_d = \tan^{-1}\left(\frac{\cos \theta_d \frac{u_x \sin \psi - u_y \cos \psi}{u_z + g}}{u_z + g}\right)$$

and the modified altitude input $U_1$ using

$$U_1 = \frac{m}{\cos \phi \cos \theta}(u_z + g)$$
Finding a suitable set of control gains for a system can be an ambiguous and time-consuming process, particularly when optimal control methods, such as Linear Quadratic Regulators, are not in use. This is especially true of highly nonlinear systems, such as quadcopters. In the previous chapter Sliding Mode Control (SMC) was chosen for use on the quadcopter test platform due to its robustness against system uncertainties and unmodeled dynamics. Stable and reasonable ranges for the SMC gains were found through Lyapunov analysis and literature review. In this chapter the method of Particle Swarm Optimization (PSO) is introduced and used to automatically tune attitude and altitude SMC gains for a quadrotor experimental test platform. These results were presented in [27].

3.1 Particle Swarm Optimization

A finite search space is established where each dimension is a parameter to be optimized. There is no limit on the number of optimization variables. As such PSO is able to optimize a system in an N-dimensional hyperspace. Artificial particles move through this search space to find an optimal location. A particle’s N-dimensional location in the search space is used in a cost function to determine that location’s fitness, a scalar value. In general optimization practice, a low cost is considered favorable. Cost functions can take many forms; the Ackley, Easom, and McCormick functions are commonly used for examining an optimization technique’s efficacy [32]. PSO iterates to minimize the cost function until an ending condition is met. The most common end condition, which is used in this study, is a set number of iterations.

Initialization - PSO begins by generating two initial test locations at random within the search space for each particle. These two points are used to generate an initial velocity from one random
point to the other for each particle in the swarm. The cost at each of these locations is calculated to provide seed data for each particle’s personal (or local) best tested location. From the swarm’s set of personal best locations, a global best location is selected. The particles’ velocity, personal best location, and the swarm’s global best location are used to determine where each particle will move next in the search space.

**PSO Equation** - The general PSO equation for the $j$-th particle in a swarm at any step $k$ is

$$
\Delta x_j(k + 1) = W(k) \Delta x_j(k) + P(k) r_1(j, k) [P_{b,j}(k) - x_i(k)] \\
+ G(k) r_2(j, k) [G_b(k) - x_j(k)]
$$

$$
x_j(k + 1) = x_j(k) + \Delta x_j(k + 1)
$$

where $\Delta x_j(k)$ and $\Delta x_j(k+1)$ are the current and next velocities, $x_j(k)$ and $x_j(k+1)$ are the current and next positions, $P_{b,j}(k)$ and $G_b(k)$ are the personal and global best locations, respectively, $r_1(j, k)$ and $r_2(j, k)$ are random values on the interval $[0,1]$ generated for each particle at each step, and $W(k)$, $P(k)$, and $G(k)$ and the inertial, personal, and global weights which can be constant or vary during the optimization process. Depending on these weights, different swarm behaviors can emerge. Without loss of generality, the inertial, personal, and global weighting functions for this study are chosen as

$$
G(k) = e^{4(k/k_f - 1)}
$$

$$
P(k) = 1 - G(k)
$$

$$
W(k) = e^{-4k/k_f}
$$

where $k_f$ is the total number of steps. These functions are visualized in Figure 3.1. By having high inertial and personal weight early in the optimization process, particles are encouraged to explore the search space, seeking out new potential solutions. As the time step $k$ increases, the personal weight decreases and the global weight becomes more significant when the optimization is 50%
of the way to the final step (solid black line in Figure 3.1). During this stage the particles tend
towards the global best solution, potential discovering new solutions due to the dominant inertial
weight. At roughly 83% completion (dashed black line in Figure 3.1), the global weight become
dominant, influencing the particles to converge to and refine a single optimal solution.

**Boundary Handling** - Several methods of ensuring particles remain in the search space were
considered including reflection, nearest, personal best, and random techniques. In [33], it was
found that the reflection boundary handling method resulted in the least sampling bias and will
be used for this study. Under this technique, when a particle’s next test position is outside of the
search space, the exterior trajectory is mirrored by the boundary resulting in a new next position
inside the search space. This is visualized in figure 3.2.

### 3.2 SMC Gain Tuning

The SMC control gains $k_i$ and $c_i$ where $i = z, \phi, \theta, \psi$ in the quadcopter control laws in
equation (2.6), reproduced below, are tuned using PSO. A set of gains take the place of $x$ in the
PSO equation (3.1) as the pseudo-location in the search space such that $x_j(k) = [k_z \ k_\phi \ k_\theta \ k_\psi \ c_z \ c_\phi
\ c_\theta \ c_\psi]^T$. The search space is restricted by the bounds given in Tables 2.2 and 2.3.
To determine the effectiveness or cost associated with each set of gains, performance metrics are calculated from step and sine wave tracking responses (without loss of generality) in each respective degree of freedom. Performance metrics include time constants, steady state error, tracking error, and controller effort. These are used to determine the cost $J_j(k)$ such that

$$J_j(k) = \sum_{i=1}^{4} (\alpha_{\tau,i}x_{i,j}(k) + \alpha_{E,i}E_{i,j}(k) + \alpha_{T,i}T_{i,j}(k) + \alpha_{W,i}W_{i,j}(k) + \alpha_{V,i}V_{i,j}(k))$$  

where
\[ y(\tau_{i,j}(k)) = 0.632y_{ss} \]
\[ E_{i,j}(k) = |y_{ss} - y_d| \]
\[ T_{i,j}(k) = \int_{0}^{t_f} |y(t) - y_d(t)|dt \]
\[ W_{i,j}(k) = \int_{0}^{t_f} |U_{i,\text{step}}|dt \]
\[ V_{i,j}(k) = \int_{0}^{t_f} |U_{i,\text{sine}}|dt \]  

(3.4)

Here \( i \) is each degree of freedom: \( z, \phi, \theta, \) and \( \psi \). \( \tau_{i,j}(k), E_{i,j}(k), \) and \( W_{i,j}(k) \) are the time constant, steady state error, and control effort, respectively, from the step input response. \( y_d \) and \( y_d(t) \) are a unit step and sine wave desired inputs, respectively. \( y_{ss} \) is the steady state value of the unit step response. Steady state is reached if the absolute value of the first derivative of the response to a step input reduces to a threshold chosen a priori before the final time of the simulation. Assuming a first order response, \( \tau_{i,j}(k) \) is then found as the time at which the step response reaches one time constant, as defined in [34]. If the first derivative does not become smaller than the threshold before the end of the simulation, it is assumed steady state is not reached and an arbitrarily large \( \tau_{i,j}(k) \) is assigned. \( T_{i,j}(k) \) and \( V_{i,j}(k) \) are the tracking error and control effort, respectively, from tracking a sine wave. Each of these values are calculated such that they are always positive. Therefore, PSO will work to minimize (3.3). The \( \alpha \) terms are combined normalizing and weighting gains.

Optimizations with this cost function are performed with simulations in both coupled and decoupled cases. In the coupled case, the simulated quadcopter receives command signals in all degrees of freedom simultaneously. For example, step response maneuvers for altitude, roll, pitch, and yaw are executed at the same time. In the decoupled case each degree of freedom is tested independently while all others are told to remain at zero. Furthermore, optimizations with various particle swarm sizes and iterations are performed and the results compared.

Other cost functions used in literature review are often more simple than equation (3.3). As such the optimization results using the above cost function will be compared to optimizations using the cost
\[ J_j(k) = \sum_{i=1}^{4} \alpha_{i,j} \text{MSE}_{i,j}(k) \]  

(3.5)

where \( \text{MSE}_{i,j}(k) \) is the mean square error of a coupled step response of each degree of freedom \( i = z, \phi, \theta, \) and \( \psi \). The resulting gains from optimizations using each cost function and simulation technique are compared for consistency of results and time require to complete the optimization process. The method which can produce consistent results in a comparatively reasonable amount of time is used to obtain controller gains for implementation on a physical platform.

### 3.3 Simulation Results and Discussion

The Particle Swarm optimization of the SMC gain was performed with various combinations of total iterations \( k_f \), numbers of particles \( N \), and the aforementioned cost functions. Sample result are shown in Tables 3.1, 3.2, and 3.3 below. Additionally, cost figures, controller gain evolution plots, and final step and sine tracking responses can be found in Appendix A.

<table>
<thead>
<tr>
<th>Table 3.1: PSO Tuned Quadcopter SMC Gains with Coupled Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain</td>
</tr>
<tr>
<td>( k_z )</td>
</tr>
<tr>
<td>( k_{\phi} )</td>
</tr>
<tr>
<td>( k_{\theta} )</td>
</tr>
<tr>
<td>( k_{\psi} )</td>
</tr>
<tr>
<td>( c_z )</td>
</tr>
<tr>
<td>( c_{\phi} )</td>
</tr>
<tr>
<td>( c_{\theta} )</td>
</tr>
<tr>
<td>( c_{\psi} )</td>
</tr>
<tr>
<td>Time</td>
</tr>
</tbody>
</table>

Through successive runs, several general observations are made. Firstly, it was seen that without a sufficient number of steps in the optimization process, particles do not converge to a solution. Conversely, allowing the optimization process to continue for a very large number of steps, such as 1,000, does not produce any noticeable improvements to a final solution. Next, the number of particles is found to greatly impact the swarm’s ability to find an optimal solution. With fewer particles in the swarm, it is less likely an adequate controller will be found. Lastly, the runtime
Table 3.2: PSO Tuned Quadcopter SMC Gains with Decoupled Responses

<table>
<thead>
<tr>
<th>Gain</th>
<th>N=10</th>
<th>N=100</th>
<th>N=30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k_f=100$</td>
<td>$k_f=10$</td>
<td>$k_f=100$</td>
</tr>
<tr>
<td>$k_z$</td>
<td>195.02</td>
<td>1.44</td>
<td>172.6</td>
</tr>
<tr>
<td>$k_{\phi}$</td>
<td>25.04</td>
<td>17.74</td>
<td>15.66</td>
</tr>
<tr>
<td>$k_{\theta}$</td>
<td>10.03</td>
<td>28.67</td>
<td>15.12</td>
</tr>
<tr>
<td>$k_{\psi}$</td>
<td>20.71</td>
<td>15.83</td>
<td>17.31</td>
</tr>
<tr>
<td>$c_z$</td>
<td>3.81</td>
<td>2.92</td>
<td>2.94</td>
</tr>
<tr>
<td>$c_{\phi}$</td>
<td>1.19</td>
<td>3.73</td>
<td>2.12</td>
</tr>
<tr>
<td>$c_{\theta}$</td>
<td>3.63</td>
<td>1.12</td>
<td>2.41</td>
</tr>
<tr>
<td>$c_{\psi}$</td>
<td>0.93</td>
<td>1.58</td>
<td>0.48</td>
</tr>
<tr>
<td>Time</td>
<td>11.77 hrs</td>
<td>12.18 hrs</td>
<td>35.1 hrs</td>
</tr>
</tbody>
</table>

Table 3.3: PSO Tuned Quadcopter SMC Gains from Coupled Responses and MSE Cost

<table>
<thead>
<tr>
<th>Gain</th>
<th>N=10</th>
<th>N=100</th>
<th>N=30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k_f=100$</td>
<td>$k_f=10$</td>
<td>$k_f=100$</td>
</tr>
<tr>
<td>$k_z$</td>
<td>42.49</td>
<td>13.96</td>
<td>33.04</td>
</tr>
<tr>
<td>$k_{\phi}$</td>
<td>21.36</td>
<td>20.14</td>
<td>22.13</td>
</tr>
<tr>
<td>$k_{\theta}$</td>
<td>14.8</td>
<td>10.95</td>
<td>19.05</td>
</tr>
<tr>
<td>$k_{\psi}$</td>
<td>20.81</td>
<td>28.89</td>
<td>22.23</td>
</tr>
<tr>
<td>$c_z$</td>
<td>2.91</td>
<td>2.63</td>
<td>3.53</td>
</tr>
<tr>
<td>$c_{\phi}$</td>
<td>3.10</td>
<td>3.08</td>
<td>2.92</td>
</tr>
<tr>
<td>$c_{\theta}$</td>
<td>1.91</td>
<td>1.54</td>
<td>1.81</td>
</tr>
<tr>
<td>$c_{\psi}$</td>
<td>2.51</td>
<td>2.11</td>
<td>2.55</td>
</tr>
<tr>
<td>Time</td>
<td>1.78 hrs</td>
<td>1.53 hrs</td>
<td>4.33 hrs</td>
</tr>
</tbody>
</table>

for the optimization process depends almost exclusively on the number of simulations performed. Thus, more steps and more particles result in longer runtime. This also applies to the coupled versus decoupled simulation responses. Because the decoupled method uses independent simulations for each degree of freedom, it takes a significantly longer amount of time to complete the optimization using this method than using coupled responses. Furthermore no significant difference in responses is seen between the two methods.

When considering the two cost functions presented in the previous section, it is found the mean square error cost function yielded a controller that does not perform as well as the controller
resulting from Equation (3.3). This is most notable in step responses where significant oscillations are present.

With these observations in mind, a balance of efficacy and efficiency is found for final tuning of the Quadcopter SMC gains. Coupled responses are used with a swarm consisting of 30 particles iterated for 100 steps using Equation (3.3) as the cost function. This setup yields consistent results. Furthermore, particles exhibit the desired behavior from the selected inertial, global, and personal weights as can be seen through the cost at each step for each particle in Figure 3.3. It is also observed that the cost has discrete values when very large. This may be an artifact of the plotting technique, a result of Sliding Mode Controller saturation, caused by cost parameter assignment when bounds are exceeded, or a combination of these or other factors. The final gains used for implementation on physical test platforms and in later simulations are given in table 3.4.

Table 3.4: Optimized SMC Gains

<table>
<thead>
<tr>
<th>Gain</th>
<th>$z$</th>
<th>$\phi$</th>
<th>$\theta$</th>
<th>$\psi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>123.7173</td>
<td>20.1745</td>
<td>19.7145</td>
<td>8.5217</td>
</tr>
<tr>
<td>$c$</td>
<td>3.4678</td>
<td>2.5823</td>
<td>2.3369</td>
<td>2.0841</td>
</tr>
</tbody>
</table>
CHAPTER 4
ARTIFICIAL POTENTIAL FIELD SWARM COORDINATION

In this chapter a multi-agent control scheme capable of aggregating multiple autonomous robotic agents into a formation as well as coordinating supplemental agents in a larger swarm is presented. The algorithm is based on the leader-follower Artificial Potential Field-Sliding Mode Control (APF-SMC) presented in [24]. Several modifications are introduced to the APF equation to achieve formation aggregation and additional analysis is performed to prove certain assumptions made in the original Lyapunov stability proof. These modifications were presented in [35]. The new swarm coordinator is then proven stable following the form of the original work.

4.1 Leader-Follower Framework

The APF presented in [24] was built on attractive and repulsive forces as function of the distances between each pairs of agenta. The attractive forces encourage swarm cohesion and aggregation while the repulsive forces prevent agents from colliding and the swarm from collapsing into a singularity. The potential field took the general form

\[
\nabla x_i V_i(X) = \sum_{j=1}^{N} f^a_{i,j}(\|x_i - x_j\|)(x_i - x_j) - f^r_{i,j}(\|x_i - x_j\|)(x_i - x_j) + f^a_{i,L}(\|x_i - x_L\|)(x_i - x_L) - f^r_{i,L}(\|x_i - x_L\|)(x_i - x_L)
\]

(4.1)

where \( V_i(X) \) is the potential at agent \( i \) as a function of the states of all agents \( X \) in the swarm with \( N \) agents, \( x \) is the position of agents \( i \) and \( j \), \( f^a \) and \( f^r \) are the attractive and repulsive force functions, respectively, and subscript \( L \) denotes the leader agent. A proposed APF function was
\[ V_i(X) = \sum_{j=1}^{N} \left( \frac{k_a}{2} \| x_i - x_j \| + \frac{k_r r}{2} e^{-\frac{\| x_i - x_j \|^2}{r}} \right) + \frac{k_a L}{2} \| x_i - x_L \| + \frac{k_r L r L}{2} e^{-\frac{\| x_i - x_L \|^2}{L}} \] (4.2)

where \( k_a \) and \( k_r \) are attractive and repulsive forces respectively, \( r \) influences the area of effect of the repulsive force, \( x_i \) and \( x_j \) are the positions of the agents \( i \) and \( j \). This is the starting point for the formulation of a decentralized swarm control algorithm with internal formation.

4.2 Formulation

4.2.1 Decentralization

By removing the leader components of equation (4.1) and separating the attractive forces from the agent dynamics and assigning them to attractive nodes \( \nu_k \), which have their own prescribed dynamics. The resulting general equation for a swarm of \( N \) agents and \( M \) attractive nodes is

\[ \nabla_{x_i} V_i(X, N) = \sum_{k=1}^{M} f_{a,i,k}^{a}(\| x_i - \nu_k \|)(x_i - \nu_k) - \sum_{j=1}^{N} f_{i,j}^{r}(\| x_i - x_j \|)(x_i - x_j) \] (4.3)

More specifically,

\[ V_i(X, N) = \sum_{k=1}^{M} k_a \| x_i - \nu_k \| + \sum_{j=1}^{N} k_r e^{-\frac{\| x_i - x_j \|^2}{r}} \] (4.4)

where \( N \) is the states of all nodes in the potential field. Attractive nodes can be stationary in space for regulation or move for tracking and can be used to aggregate the swarm into a specific formation or structure. The repulsive forces remain dependent on agent dynamics to avoid collision.

4.2.2 Balancing Forces

During preliminary simulations using APF (4.4) to control a swarm of double-integrator point masses, it was found that a zero-gradient region would form in the area enclosed by attractive nodes due to symmetry in the attractive forces. An example of this can be seen in Figure 4.1a. Furthermore, when more agents than nodes were in a swarm, often times no agent would reach the node. This can be seen in Figure 4.1b. It was found that by introducing counteractive or
balancing forces of a lower magnitude than the attractive and repulsive forces already present, these effects could be mitigated. That is, introducing a small linear repulsive force has the effect of pushing agents towards nodes when in the aforementioned zero-gradient region. Introducing a small Gaussian attractive force or "sink" holds agents at a node. With the inclusion of these forces the APF becomes

$$V_i(X, N) = \sum_{k=1}^{M} \left( k_a \|x_i - \nu_k\| - k_s e^{-\|x_i - \nu_k\|^2/a} \right) + \sum_{j=1}^{N} \left( k_r e^{-\|x_i - x_j\|^2/r} - k_l \|x_i - x_j\| \right)$$

(4.5)

where $k_l$ is the long range repulsive gain, $k_s$ is the short range attractive gain, and $a$ influences the area of effect of the short range attractive force.

4.2.3 Smoothness

To avoid singularities and potentially unstable dynamics, the APF must be smooth (i.e. continuously differentiable). In Equation (4.5), linear terms with gains $k_a$ and $k_l$ result in discontinuities
in the gradient when $\nu_k$ or $x_j$ is equal to $x_i$. As such, these terms are modified with a hyperbolic function with a "smoothness" factor $c_m$. The final form of the proposed APF equation is

$$V_i(\mathbf{X}, N) = \sum_{k=1}^{M} \left( k_a \sqrt{\|x_i - \nu_k\|^2 + c_m^2} - k_s e^{-\|x_i - \nu_k\|^2} / a \right) + \sum_{j=1}^{N} \left( k_r e^{-\frac{\|x_i - x_j\|^2}{r}} - k_l \sqrt{\|x_i - x_j\|^2 + c_m^2} \right)$$

(4.6)

The profile of this APF in one dimension with one agent and one attractive node, both at the origin, is shown in Figure 4.2. the attractive forces $f^a$ and repulsive forces $f^r$ for the APF in the form of Equation (4.3), are

$$f^a_{i,k}(\|x_i - \nu_k\|) = \frac{k_a}{\sqrt{\|x_i - \nu_k\|^2 + c_m^2}} + \frac{2k_s e^{-\|x_i - \nu_k\|^2}}{a}$$

(4.7)

$$f^r_{i,j}(\|x_i - x_j\|) = \frac{2k_r e^{-\frac{\|x_i - x_j\|^2}{r}}}{r} + \frac{k_l}{\sqrt{\|x_i - x_j\|^2 + c_m^2}}$$

(4.8)

respectively.
4.3 Equilibrium Distance

In Fabian et al. [24] it is assumed that there exists an equilibrium distance $d_e$ where agents will settle in relation to each other and the attractive and repulsive forces are equal. When considering the overall swarm behavior it is useful to be able to assign an equilibrium distance such that agents will settle with safe and proper space between themselves. As such, Equation (4.6) will be analyzed to find the analytical solution to the equilibrium distance as well as determine bounds and expressions for gains to guarantee the specified equilibrium distance.

**Definition 4.3.1 (Equilibrium distance).** There equilibrium distance $d_e$ is a distance away from an agent where the attractive and repulsive forces balance each other satisfying $f_{a_{i,k}}(d_e) = f_{r_{i,k}}(d_e)$. Moreover, $f_{a_{i,k}}(d) > f_{r_{i,j}}(d)$ for all $d > d_e$ and $f_{a_{i,k}}(d) < f_{r_{i,j}}(d)$ for all $0 < d < d_e$.

**Definition 4.3.2 (Lambert-W Function).** Also called the product logarithm, the Lambert-W function $W(z)$ is the inverse function of $f(z) = z e^z$. That is,

\[
\text{If } f(z) = z e^z \\
\text{Then } W(z) = f^{-1}(z) \tag{4.9}
\]

Useful properties of the Lambert-W function used in the following analysis include

\[
W(ze^z) = z \tag{4.10}
\]

\[
W(z)e^{W(z)} = z \tag{4.11}
\]

Furthermore, for inputs $x \in \mathbb{R}$, the real output of $W(x)$ is divided into the principal branch $W_0$ and the lower branch $W_{-1}$ plotted in Figure 4.3 [36].

**Assumption 4.3.1.** A swarm governed by (4.6) will settle such that $\bar{x} = \bar{\nu}$ and that agents and nodes are close to the centroid; $x_j \approx \bar{x}$ and $\nu_k \approx \bar{\nu}$.
Lemma 4.3.1. Consider a swarm of size $N$ agents and $M$ attractive nodes, with interactions governed by (4.6) which satisfies Assumption 4.3.1. There exists an equilibrium distance

$$d_e = \left[ \frac{-r}{2} W_{-1} \left( \frac{-r}{2} K^2 e^{-\frac{2c^2}{m}} \right) - c^2_m \right]^{\frac{1}{2}} \quad (4.12)$$

and it is a local minimum in the potential field.

Proof. Applying Assumption 4.3.1 to Equation (4.6) and recognizing summation limits the APF at agent $i$ becomes

$$V_i(\mathbf{X}, N) = M \left( k_a \sqrt{\|x_i - \bar{x}\|^2 + c^2_m} - k_s e^{\frac{\|x_i - \bar{x}\|^2}{a}} \right) + (N - 1) \left( k_r e^{\frac{\|x_i - \bar{x}\|^2}{r}} - k_l \sqrt{\|x_i - \bar{x}\|^2 + c^2_m} \right) \quad (4.13)$$

Since an agent is at zero distance from itself, it cannot experience any repulsive forces from itself. Therefore, the inter-agent forces sum $N - 1$ times for a swarm of size $N$.

Assigning the balancing force gains as $k_l = \delta_l k_a$, $k_s = \delta_s k_r$ and $a = r$, like functions can now be grouped together to yield

$$V_i(\mathbf{X}, N) = [M - (N - 1)\delta_l] k_a \sqrt{\|x_i - \bar{x}\|^2 + c^2_m} + [(N - 1) - M\delta_s] k_s e^{\frac{\|x_i - \bar{x}\|^2}{a}} \quad (4.14)$$
The equilibrium point \(\|x_i - \bar{x}\| = d_e\) occurs at the local minimum of the potential field such that

\[
\frac{d}{dx_i} V_i(X, N) \bigg|_{\|x_i - \bar{x}\| = d_e} = [M - (N - 1)\delta_l] k_0 \frac{d_e}{\sqrt{d_e^2 + c_m^2}} \\
+ [(N - 1) - M\delta_s] k_r \left(\frac{-2d_e}{r}\right) e^{-\frac{d_e^2}{r}} = 0
\] (4.15)

The equilibrium distance \(d_e\) can be solved for by first manipulating (4.15) into the form \(f(d_e)e^{f(d_e)}\) such that

\[
\frac{[M - (N - 1)\delta_l] k_0 d_e}{\sqrt{d_e^2 + c_m^2}} = \frac{[(N - 1) - M\delta_s] k_0}{[(N - 1) - M\delta_s] k_r} \left(\frac{2d_e}{r}\right) e^{-\frac{d_e^2}{r}}
\]

Let \(K\) be a value such that

\[
K = \frac{[M - (N - 1)\delta_l] k_0}{[(N - 1) - M\delta_s] k_r}
\] (4.16)

Then,

\[
\frac{r}{2} K^2 e^{-\frac{2c_m^2}{r}} = \frac{2}{r} \left(\frac{d_e^2 + c_m^2}{r}\right) e^{-\frac{2d_e^2}{r}} - \frac{r}{2} \frac{2}{r} \left(\frac{d_e^2 + c_m^2}{r}\right) e^{-\frac{2c_m^2}{r}}
\]

Applying the Lambert-W function to both sides and recognizing the property Equation (4.10) yields

\[
W \left(\frac{-r}{2} K^2 e^{-\frac{2c_m^2}{r}}\right) = -2 \left(\frac{d_e^2 + c_m^2}{r}\right)
\]

\[
\frac{-r}{2} W \left(\frac{-r}{2} K^2 e^{-\frac{2c_m^2}{r}}\right) - c_m = d_e^2
\] (4.18)
The equilibrium distance becomes

\[ d_e = \left[ \frac{-r}{2} W \left( \frac{-r}{2} K^2 e^{-\frac{2c^2_m}{r}} \right) - \frac{c^2_m}{z^2} \right]^{\frac{1}{2}} \] (4.19)

Note that for \( d_e \in \mathbb{R}^+ \), \( W \left( \frac{-r}{2} K^2 e^{-\frac{2c^2_m}{r}} \right) \) must be negative and \( \frac{-r}{2} W \left( \frac{-r}{2} K^2 e^{-\frac{2c^2_m}{r}} \right) - \frac{c^2_m}{z^2} \) must be greater than zero. For Equation (4.19) to be a minimum, the APF described by Equation (4.6) must have a positive second derivative (i.e., be concave up) at \( d_e \).

Let \( \mu = \frac{-r}{2} K^2 e^{-\frac{2c^2_m}{r}} \) and \( z^2 = \frac{-r}{2} W(\mu) \), then Equation (4.19) may be written as

\[ d_e^2 = z^2 - \frac{c^2_m}{z^2} \] (4.20)

Taking the second derivative of Equation (4.14), evaluating it at \( \| x_i - \bar{x} \| = d_e \), substituting in Equation (4.20), and setting it greater than zero yields

\[
\begin{align*}
&M - (N - 1) \delta_l \left[ \frac{z - \frac{z^2 - c^2_m}{z^2}}{z^2} \right] \\
&\quad + [(N - 1) - M \delta_s] k_e \frac{-2}{r} + \frac{4}{r^2} (z^2 - c^2_m) > 0
\end{align*}
\] (4.21)

Simplifying

\[
\begin{align*}
K \left[ \frac{1}{z} - \frac{z^2 - c^2_m}{z^3} \right] &> \frac{2}{r} e^{\frac{z^2}{r}} e^{\frac{c^2_m}{r}} \left[ 1 + W(\mu) + \frac{2c^2_m}{r} \right] \\
K \frac{c^2_m}{z^2} &> \frac{2}{r} e^{W(\mu)} \frac{1}{z^3} e^{\frac{c^2_m}{r}} \left[ 1 + W(\mu) + \frac{2c^2_m}{r} \right] \\
K \frac{c^2_m}{z^2} &> \frac{2}{r} \left[ \frac{\mu}{W(\mu)} \right]^{\frac{1}{2}} e^{\frac{c^2_m}{r}} \left[ 1 + W(\mu) + \frac{2c^2_m}{r} \right] \\
K \frac{c^2_m}{z^2} &> \frac{2}{r} \left[ \frac{r^2}{4} K^2 e^{-\frac{2c^2_m}{r}} \right]^{\frac{1}{2}} e^{\frac{c^2_m}{r}} \left[ 1 + W(\mu) + \frac{2c^2_m}{r} \right] \\
\frac{c^2_m}{z^2} &> \frac{1}{r} \left[ 1 + W(\mu) + \frac{2c^2_m}{r} \right] \\
-\frac{2c^2_m}{r} &< W(\mu) \left[ 1 + W(\mu) + \frac{2c^2_m}{r} \right]
\end{align*}
\]
\[-\frac{2c_m^2}{r} < W^2(\mu) + \left(1 + \frac{2c_m^2}{r}\right) W(\mu) \quad (4.22)\]

Completing the square

\[-\frac{2c_m^2}{r} + \frac{1}{4} \left(1 + \frac{2c_m^2}{r}\right)^2 < \left[W(\mu) + \frac{1}{2} \left(1 + \frac{2c_m^2}{r}\right)\right]^2\]
\[\frac{1}{4} \left(1 - \frac{2c_m^2}{r}\right)^2 < \left[W(\mu) + \frac{1}{2} \left(1 + \frac{2c_m^2}{r}\right)\right]^2\]
\[\frac{1}{2} \left|1 - \frac{2c_m^2}{r}\right| > \left|W(\mu) + \frac{1}{2} \left(1 + \frac{2c_m^2}{r}\right)\right| \quad (4.23)\]

Considering all possible sign combinations resulting from the absolute value functions, to satisfy the inequality

\[-\frac{r}{2} W(\mu) - c_m^2 > 0 \quad (4.24)\]

necessary for a real positive equilibrium distance \(d_e\) from Equation (4.19), it is found that \(W(\mu) < -1\) must be hold true. This is satisfied using the lower branch of the Lambert-W function \(W_{-1}(z)\) for \(z \in (-\frac{1}{e}, 0)\). Therefore, the equilibrium distance

\[d_e = \left[\frac{-r}{2} W_{-1} \left(\frac{-r}{2} K^2 e^{-\frac{-2c_m^2}{r}}\right) - c_m^2\right]^\frac{1}{2} \quad (4.12)\]

is a local minimum of the potential field.

**Lemma 4.3.2.** Consider a swarm of size \(N\) agents and \(M\) attractive nodes, with interactions governed by Equation (4.6) which satisfies Assumption 4.3.1. By selecting APF parameters such that \(d_e > 0\), \(k_a > 0\), \(c_m > 0\), \(\delta_l < \frac{M}{N-1}\), \(\delta_s < \frac{N-1}{M}\), \(c_s > 1\), and calculated gains

\[k_r = k_a \left[\frac{M - (N - 1)\delta_l}{(N - 1) - M\delta_s}\right] \left(\frac{ec_m c_m^2}{\gamma e^\gamma}\right)^\frac{1}{2} \quad (4.25)\]
Where

\[
\gamma = \frac{-c_m^2}{d_e^2 + c_m^2} W_{-1}\left(\frac{-1}{ec_s}\right)
\]  \hspace{1cm} (4.26)

\[
r = \frac{2c_m^2}{W_0 \left(K^2ec_s c_m^2\right)}
\]  \hspace{1cm} (4.27)

Here,

\[
K = \left[ M - (N - 1)\delta_l \right] \frac{k_a}{\left[ (N - 1) - M\delta_s \right] k_r}
\]  \hspace{1cm} (4.16)

The resulting artificial potential field will have a local minimum at a distance \(d_e\) away from an agent.

**Proof.** For real \(d_e\) in Equation (4.12)

\[
-\frac{r}{2} W_{-1}\left(\frac{-r}{2} K^2 e^{-\frac{2c_m^2}{r}}\right) - c_m^2 > 0
\]

\[
W_{-1}\left(\frac{-r}{2} K^2 e^{-\frac{2c_m^2}{r}}\right) < -\frac{2c_m^2}{r}
\]  \hspace{1cm} (4.28)

Note that \(W_{-1}(z) \in \mathbb{R}^-\) for \(-1/e < z < 0\) only. Therefore,

\[
\frac{-1}{e} < -\frac{r}{2} K^2 e^{-\frac{2c_m^2}{r}} < 0
\]

\[
0 < \frac{r}{2} K^2 e^{-\frac{2c_m^2}{r}} < \frac{1}{e}
\]  \hspace{1cm} (4.29)

The design parameter \(c_s > 1\) is introduced such that

\[
\frac{r}{2} K^2 e^{-\frac{2c_m^2}{r}} = \frac{1}{ec_s}
\]  \hspace{1cm} (4.30)
satisfies the inequality (4.29). Equation (4.30) may be used to solve for the APF gain \( r \) as follows:

\[
\frac{r^2}{2 e^{-2 r^2 / r}} = \frac{1}{K^2 e c_s} \\
2 e^{-2 r^2 / r} = K^2 e c_s \\
2 c_m^2 e^{-2 r^2 / r} = K^2 e c_s c_m^2 \\
2 c_m^2 / r = W_0 \left( K^2 e c_s c_m^2 \right) \\
r = \frac{2 c_m^2}{W_0 \left( K^2 e c_s c_m^2 \right)}
\] (4.27)

To solve for one of the gains, Equations (4.30) and (4.27) are substituted into Equation (4.12) yielding

\[
d_e = \left[ \frac{-c_m^2}{W_0 \left( K^2 e c_s c_m^2 \right)} W_{-1} \left( \frac{-1}{e c_s} \right) - c_m^2 \right]^{\frac{1}{2}} \\
\frac{-c_m^2}{d_e^2 + c_m^2} = \frac{W_{-1} \left( \frac{-1}{e c_s} \right)}{W_0 \left( K^2 e c_s c_m^2 \right)} \\
W_0 \left( K^2 e c_s c_m^2 \right) = \frac{-c_m^2}{d_e^2 + c_m^2} W_{-1} \left( \frac{-1}{e c_s} \right)
\] (4.31)

Define \( \gamma \) such that

\[
\gamma = \frac{-c_m^2}{d_e^2 + c_m^2} W_{-1} \left( \frac{-1}{e c_s} \right)
\] (4.26)

Then

\[
W_0 \left( K^2 e c_s c_m^2 \right) = \gamma
\] (4.32)

Applying the properties of the Lambert-W function yields

\[
K^2 e c_s c_m^2 = \gamma e^\gamma \\
K^2 = \frac{\gamma e^\gamma}{e c_s c_m^2}
\]
Substituting Equation (4.16) yields

\[
\frac{[M - (N - 1)\delta_l]}{[(N - 1) - M\delta_s]} k_a = \left(\frac{\gamma e^\gamma}{e c_s c_m^2}\right)^{\frac{1}{2}} k_r
\]

Thus, the repulsive gain \(k_r\) can be defined such that

\[
k_r = k_a \frac{[M - (N - 1)\delta_l]}{[(N - 1) - M\delta_s]} \left(\frac{e c_s c_m^2}{\gamma e^\gamma}\right)^{\frac{1}{2}}
\]  

(4.25)

Noting that \(k_a \in \mathbb{R}^+\),

\[
M - (N - 1)\delta_l > 0 \\
\delta_l < \frac{M}{N - 1}
\]

(4.33)

and

\[
(N - 1) - M\delta_s > 0 \\
\delta_s < \frac{N - 1}{M}
\]

(4.34)

The inequalities (4.33) and (4.34) must hold to yield a finite, real, \(k_r\). Thus, selecting APF parameters such that \(d_e > 0, k_a > 0, c_m > 0, \delta_l < \frac{M}{N - 1}, \delta_s < \frac{N - 1}{M}, c_s > 1,\) and gains \(k_r\) and \(r\) from Equations (4.25) and (4.27), respectively, results in an Artificial Potential Field with a local minimum at a distance \(d_e\) away from an agent.

4.4 Stability Under First-Order Dynamics

4.4.1 Swarm Centroid Dynamics

This stability analysis follows the process developed and presented by Fabian et al. in [24]. Consider the first order dynamics and control input

\[
\dot{x}_i = u_i = -\nabla x_i V_i(X, N)
\]

(4.35)
and the APF

\[ \nabla_{x_i} V_i(X, N) = \sum_{k=1}^{M} f^a_{i,k}(\|x_i - \nu_k\|) (x_i - \nu_k) - \sum_{j=1}^{N} f^r_{i,j}(\|x_i - x_j\|) (x_i - x_j) \quad (4.3) \]

with gains selected via Lemma 4.3.2. Node dynamics are assigned as

\[ \dot{\nu}_k = \mu_k(t) \quad (4.36) \]

which is chosen by the design engineer. The centroid of the attractive nodes is

\[ \bar{\nu} = \frac{1}{M} \sum_{k=1}^{M} \nu_k \quad (4.37) \]

**Assumption 4.4.1.** Let \( f^a(t) = f^a_{i,k}(\|x_i - \nu_k\|) \). There exist \( F^a, F^a \in \mathbb{R}^+ \) such that

\[ F^a \leq f^a(t) \leq F^a \quad (4.38) \]

for all \( t \). Furthermore, there exists \( F^r \in \mathbb{R}^+ \) such that

\[ 0 < f^r_{i,j}(\|x_i - x_j\|) \|x_i - x_j\| < F^r \quad (4.39) \]

**Lemma 4.4.1.** Consider a swarm of size \( N \) agents with dynamics described in Equation (4.35), of \( M \) attractive nodes with dynamics of Equation (4.36), with interactions governed by (4.6) and with gains selected via Lemma 4.3.2. The swarm centroid dynamics are governed exclusively by attractive node interaction forces such that the swarm centroid \( \bar{x} \) dynamics follow

\[ \dot{\bar{x}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{M} f^a_{i,k}(\|x_i - \nu_k\|) (x_i - \nu_k) \quad (4.40) \]
Proof. The swarm centroid $\bar{x}$ is defined as

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$ (4.41)

Differentiating and applying the control law in Equation (4.35) yields

$$\dot{\bar{x}} = \frac{1}{N} \sum_{i=1}^{N} \dot{x}_i = \frac{1}{N} \sum_{i=1}^{N} u_i$$ (4.42)

$$\dot{\bar{x}} = \frac{-1}{N} \sum_{i=1}^{N} \left[ \sum_{k=1}^{M} f_{i,k}^{a}(\|x_i - \nu_k\|)(x_i - \nu_k) - \sum_{j=1}^{N} f_{i,j}^{r}(\|x_i - x_j\|)(x_i - x_j) \right]$$ (4.43)

Note that inter-agent forces are equal and opposite for any given agent pair. That is, $f_{i,j}^{r}(\|x_i - x_j\|) = -f_{j,i}^{r}(\|x_j - x_i\|)$. Therefore, the second term of Equation (4.43) is zero, resulting in Equation (4.40).

By applying Assumption 4.4.1 and recognizing the summation bounds to Equation (4.40), the swarm centroid dynamics can be reduced to

$$\dot{\bar{x}} = -M f^{a}(t)(\bar{x} - \bar{\nu})$$ (4.44)

Lemma 4.4.2. Consider a swarm of size $N$ agents with dynamics described in Equation (4.35), of $M$ attractive nodes with dynamics of Equation (4.36), with interactions governed by (4.6) and with gains selected via Lemma 4.3.2. As $t \to \infty$ the swarm centroid $\bar{x}$ will asymptotically approach a hyperball around node centroid $\bar{\nu}$ of radius $\bar{\epsilon}$ such that

$$\bar{\epsilon} = \frac{\bar{\mu}(t)}{M f^{a}(t)}$$ (4.45)

where $\bar{\mu}(t)$ is the average movement of the attractive nodes.
Proof. Consider the Lyapunov candidate

\[ L(x) = \frac{1}{2} \tilde{e}^T \tilde{e} \quad (4.46) \]

where \( \tilde{e} = \bar{x} - \bar{\nu} \) represent the vector from the agent swarm centroid \( \bar{x} \) to the node centroid \( \bar{\nu} \).

Taking the time derivative of the Lyapunov candidate yields

\[ \dot{L} = \dot{\tilde{e}}^T \tilde{e} = (\dot{\bar{x}} - \dot{\bar{\nu}})^T (\bar{x} - \bar{\nu}) = \dot{\bar{x}}^T \tilde{e} - \dot{\bar{\nu}}^T \tilde{e} \quad (4.47) \]

By taking the time derivative of Equation (4.37) and applying Equation (4.36) we obtain the average movement of the nodes as

\[ \dot{\bar{\nu}} = \frac{1}{M} \sum_{k=1}^{M} \mu_k(t) = \bar{\mu}(t) \quad (4.48) \]

By applying Lemma 4.4.1 and Assumption 4.4.1 yields

\[ \dot{L} = -M f^a(t)(\bar{x} - \bar{\nu})^T \tilde{e} - \bar{\mu}(t)^T \tilde{e} \]
\[ \dot{L} \leq -M f^a(t)\| \tilde{e} \|^2 + \bar{\mu}(t)\| \tilde{e} \| \quad (4.49) \]

For stability and aggregation \( \dot{L} \leq 0 \). Therefore

\[ -M f^a(t)\| \tilde{e} \| + \bar{\mu}(t) < 0 \]
\[ \| \tilde{e} \| > \frac{\bar{\mu}(t)}{M f^a(t)} \equiv \tilde{\epsilon} \quad (4.50) \]

Thus, while the above inequality holds, the swarm centroid will converge to the node centroid. \( \square \)

Remark 4.4.1. Rearranging Equation (4.50) as

\[ \bar{\mu}(t) < M f^a(t)\| \tilde{e} \| \quad (4.51) \]

and recall that \( f^a(t) = f^a_{i,k}(\| x_i - \nu_k \|) \), \( \tilde{e} = \bar{x} - \bar{\nu} \), and the APF gradient in the form of Equation (4.3), the right hand side of Equation (4.51) can be interpreted as the total average attractive force.
experienced by agents in the swarm. Therefore, from Equation (4.51), to maintain stability and achieve swarm aggregation, the average node movement must be less than the total average attractive force. Moreover, if the nodes are stationary (i.e., $\mu_k(t) = 0$ for all $k = 1, 2, \ldots, M$), the swarm centroid $\bar{x}$ will converge to the node centroid $\bar{\nu}$.

### 4.4.2 Agent Dynamics

**Theorem 4.4.3.** Consider a swarm of size $N$ agents with dynamics described in Equation (4.35), of $M$ attractive nodes with dynamics of Equation (4.36), with interactions governed by (4.6) and with gains selected via Lemma 4.3.2. The agents will converge to a hyperball around the swarm centroid of radius

$$
\epsilon = \frac{(N - 1)F^r}{MF^{fa}}
$$

**Proof.** Let $e_i = x_i - \bar{x}$, and consider the Lyapunov function $L_i = \frac{1}{2}e_i^T e_i$. Taking the time derivative of the Lyapunov candidate along agent trajectories yields

$$
\dot{L}_i = e_i^T \dot{e}_i = (\dot{x}_i - \dot{\bar{x}})^T e_i = (u_i - \dot{\bar{x}})^T e_i
$$

Applying Assumption 4.4.1 and recognizing summation limits yields

$$
\dot{L}_i = \Theta_1 + \Theta_2 + \Theta_3
$$

$$
\Theta_1 = \sum_{j=1}^{N} f_{i,j}^r(\| x_i - x_j \|)(x_i - x_j)^T e_i
$$

$$
\Theta_2 = -\sum_{k=1}^{M} f_{i,k}^a(\| x_i - \nu_k \|)(x_i - \nu_k)^T e_i
$$

$$
\Theta_3 = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{M} f_{i,k}^a(\| x_i - \nu_k \|)(x_i - \nu_k)^T e_i
$$

Applying Assumption 4.4.1 and recognizing summation limits yields
\[ \Theta_1 = (N - 1)F^r \| e_i \| \]
\[ \Theta_2 = -M f^a(t)(x_i - \nu_k)^T e_i \]
\[ = -M f^a(t)(e_i + \bar{e})^T e_i \]
\[ \Theta_3 = M f^a(t)(\bar{x} - \bar{\nu}) \]
\[ = M f^a(t)\bar{e}^T e_i \]

Thus,
\[
\dot{L}_i \leq (N - 1)F^r \| e_i \| - M f^a(t)(e_i + \bar{e})^T e_i + M f^a(t)\bar{e}^T e_i \\
\leq (N - 1)F^r \| e_i \| - M f^a(t)e_i^T e_i \\
\leq (N - 1)F^r \| e_i \| - M f^a(t)\| e_i \|^2 \\
\leq \| e_i \| \left[ (N - 1)F^r - M f^a(t)\| e_i \| \right] \tag{4.55}
\]

Since \( \| e_i \| \geq 0 \), the time derivative of the Lyapunov candidate is negative if
\[
\| e_i \| > \frac{(N - 1)F^r}{M f^a(t)} \equiv \epsilon \tag{4.56}
\]

Thus, while \( \| e_i \| > \epsilon \), agents will be attracted to the set point \( e_i = x_i - \bar{x} = 0 \) and converge asymptotically to the hypersphere of radius \( \epsilon \) around the swarm centroid \( \bar{x} \).

**Corollary 4.4.3.1.** Given proper APF gain selection, while the distances from an agent to each attractive node \( \| x_i - \nu_k \| \) and the distances between agents \( \| x_i - x_j \| \) are greater than the equilibrium distance \( d_e \), agents will converge towards the swarm centroid \( \bar{x} \).

**Proof.** Consider Theorem 4.4.3 at Equation (4.54). By recognizing summation limits, applying Assumption 4.4.1 and similarly assuming there exists a function \( f^r(t) \) where \( f^r(t) = f^r_{i,j}(\| x_i - x_j \|) \), one obtains
\[ \dot{L}_i \leq (N-1)f_r(t)e_i^T e_i - Mf^a(t)(e_i + \bar{e})^T + Mf^a(t)\bar{e}^T e_i \]
\[ \leq [(N-1)f_r(t) - Mf^a(t)]\|e_i\|^2 \]

(4.57)

Thus Lyapunov stability is achieved when

\[ [(N-1)f_r(t) - Mf^a(t)] < 0 \]
\[ (N-1)f_r(t) < Mf^a(t) \]  

(4.58)

Considering the state dependent force Equations \( f_{i,k}^a(||x_i - \nu_k||) \) and \( f_{i,j}^r(||x_i - x_j||) \) in Equation (4.58), and recalling that using gains determined via Lemma 4.3.2 satisfy Lemma 4.3.1 it is found that while \( ||x_i - \nu_k|| \) and \( ||x_i - x_j|| \) are greater than \( d_e \), Equation (4.58) is satisfied and the controller is stable in the sense of Lyapunov for \( e_i = x_i - \bar{x} \rightarrow 0 \) as \( t \rightarrow \infty \).

From Lemmas 4.4.1 and 4.4.2 and Theorem 4.4.3, it is proven that agents converge to a hyperball around the swarm centroid \( \bar{x} \), which converges to the centroid of the attractive nodes. Therefore, the agents converge to a hyperball around the node centroid.

### 4.5 Stability Under General Dynamics

With the first order system and input described in Equation (4.35) shown to be stable, these results can be extended to general higher order systems using Sliding Mode Control. This analysis is based on [24] and, as such, requires the same assumptions and has the same process as follows. Consider general agent dynamics governed by

\[ M_i(x_i)\ddot{x}_i + g_i(x_i, \dot{x}_i) = u_i \]

(4.59)

where \( M_i(x_i) \) is the inertial matrix for agent \( i \) and \( g_i \) contains any higher order or nonlinear dynamics.
Assumption 4.5.1. For all agents \( i = 1, 2, ..., N \)

\[
\frac{M_i}{M_i} \|y\|^2 \leq y^T M_i(x_i)y \leq \overline{M}_i \|y\|^2 < \infty
\]  

(4.60)

for real scalars

\[
0 < M_i < \overline{M}_i < \infty
\]

(4.61)

and any arbitrary \( y \)

Assumption 4.5.2. For all \( t \geq 0 \) the time derivatives of the force functions must be such that

\[
\dot{f}_{i,k}^a(\|x_i - \nu_k\|)\|x_i - \nu_k\| \leq \Gamma^a(X, N) \text{ for all } i, k
\]

\[
\dot{f}_{i,j}^r(\|x_i - x_j\|)\|x_i - x_j\| \leq \Gamma^r(X) \text{ for all } i, j
\]

(4.62)

where \( \Gamma^a \) and \( \Gamma^r \) are state dependent upper bounds, however, global bounds are sufficient.

Assumption 4.5.3. For all \( \{i,j\} \) and \( \{i,k\} \) pairs, where \( i = 1, 2, ..., N, j = 1, 2, ..., N, \) and \( k = 1, 2, ..., M, \) the potential field gradients are bounded by

\[
\|\nabla_{x_i} V(X, N)\| \leq \alpha(X, N)
\]

\[
\|\nabla_{x_j}\nabla_{x_i} V(X, N)\| \leq \beta(X, N)
\]

\[
\|\nabla_{\nu_k}\nabla_{x_i} V(X, N)\| \leq \gamma(X, N)
\]

(4.63)

where \( \alpha(X, N), \beta(X, N), \) and \( \gamma(X, N) \) are known, finite functions of the agent and node states.

Assumption 4.5.4. \( \dot{x}_i(0) = 0 \) for \( i = 1, 2, ..., N \)

Theorem 4.5.1. Consider a swarm of size \( N \) agents with dynamics described by Equation (4.59), \( M \) nodes with dynamics described by Equation (4.36), with interactions governed by the Artificial
Potential Field described by Equation (4.6), gains selected via Lemma 4.3.2, and which satisfies the aforementioned assumptions. Let the control input for agent $i$ be

$$u_i = -K_i(X, N)\text{sign}(s_i) + g_i(x, x_i)$$

with gain

$$K_i(X, N) > M_i(x_i)(\overline{J}_i(X, N) + \epsilon_i)$$

for some $\epsilon > 0$, where

$$\overline{J}_i(X, N) = M \gamma(X, N)(\alpha(X(0), N(0)) + \alpha(X, N))$$
$$+ N \beta(X, N)(\alpha(X(0), N(0)) + \alpha(X, N))$$
$$+ M \Gamma^a(X, N) - (N - 1)\Gamma^r(X)$$

and sliding manifold

$$s_i = \dot{x}_i + \nabla x_i V(X, N)$$

Then sliding mode occurs in all $s_i$ and

$$\dot{x}_i = -\nabla x_i V(X, N)$$

is satisfied in finite time.

**Proof.** For the Lyapunov candidate

$$\Lambda_i = \frac{1}{2} s_i^T s_i$$

sliding mode occurs in finite time if

$$\dot{\Lambda}_i = s_i^T \dot{s}_i < -\epsilon_i \|s_i\|$$
for some $\epsilon_i > 0$, also know as the reaching condition. Taking the time derivative of the sliding manifold yields

$$
\dot{s}_i = \ddot{x}_i + \frac{d}{dt} \nabla_{x_i} V(X, N)
= M_i(x_i)^{-1} u_i - M_i(x_i)^{-1} g_i(x_i, \dot{x}_i) + \frac{d}{dt} \nabla_{x_i} V(X, N)
$$

(4.71)

Thus,

$$
\dot{s}_i^T \dot{s}_i = s_i^T \left[ M_i(x_i)^{-1} u_i - M_i(x_i)^{-1} g_i(x_i, \dot{x}_i) + \frac{d}{dt} \nabla_{x_i} V(X, N) \right]
$$

(4.72)

Recalling that

$$
\nabla_{x_i} V_i(X, N) = \sum_{k=1}^{M} f_{i,k}^a (\|x_i - \nu_k\|)(x_i - \nu_k) - \sum_{j=1}^{N} f_{i,j}^r (\|x_i - x_j\|)(x_i - x_j)
$$

(4.73)

and taking the time derivative and applying the product rule yields

$$
\frac{d}{dt} \nabla_{x_i} V_i(X, N) = \sum_{k=1}^{M} \nabla_{\nu_k} \left[ \nabla_{x_i} V(X, N) \right] \dot{\nu}_k + \sum_{j=1}^{N} \nabla_{x_j} \left[ \nabla_{x_i} V(X, N) \right] \dot{x}_j
+ \sum_{k=1}^{M} \dot{f}_{i,k}^a (\|x_i - \nu_k\|)(x_i - \nu_k) - \sum_{j=1}^{N} \dot{f}_{i,j}^r (\|x_i - x_j\|)(x_i - x_j)
$$

(4.74)

(4.75)

As stated in [24], it is shown in [37] that

$$
\left\| \sum_{k=1}^{M} \nabla_{\nu_k} \left[ \nabla_{x_i} V(X, N) \right] \dot{\nu}_k \right\| \leq M \gamma(X, N)(\alpha(X(0), N(0)) + \alpha(X, N))
$$

$$
\left\| \sum_{j=1}^{N} \nabla_{x_j} \left[ \nabla_{x_i} V(X, N) \right] \dot{x}_j \right\| \leq N \beta(X, N)(\alpha(X(0), N(0)) + \alpha(X, N))
$$

(4.76)
Therefore,
\[
\| \frac{d}{dt} \nabla x_i V(X, N) \| \leq M \gamma(X, N)(\alpha(X(0), N(0)) + \alpha(X, N)) + N \beta(X, N)(\alpha(X(0), N(0)) + \alpha(X, N)) + M \Gamma^a(X, N) - (N - 1) \Gamma^r(X)
\] (4.77)

and
\[
\| \frac{d}{dt} \nabla x_i V(X, N) \| \leq \overline{J}_i(X, N)
\] (4.78)

Then

\[
\dot{\Lambda}_i \leq s_i^T \left[ M_i(x_i)^{-1} u_i - M_i(x_i)^{-1} g_i(x_i, \dot{x}_i) + \overline{J}_i(X, N) \right]
\leq s_i^T \left[ M_i(x_i)^{-1} (-K_i(X, N) \text{sign}(s_i) + g_i(x, x_i)) - M_i(x_i)^{-1} g_i(x_i, \dot{x}_i) + \overline{J}_i(X, N) \right]
\leq s_i^T \left[ -M_i(x_i)^{-1} K_i(X, N) \text{sign}(s_i) + \overline{J}_i(X, N) \right]
\] (4.79)

Let \( K_i(X, N) \) be such that
\[
K_i(X, N) > \overline{M}_i(x_i)(\overline{J}_i(X, N) + \epsilon_i)
\] (4.80)

Then

\[
\dot{\Lambda}_i \leq -s_i^T \left[ (\overline{J}_i(X, N) + \epsilon_i) \text{sign}(s_i) - \overline{J}_i(X, N) \right]
\leq -\|s_i\|(\overline{J}_i(X, N) + \epsilon_i) - s_i^T \overline{J}_i(X, N)
\leq -\|s_i\|(\overline{J}_i(X, N) + \epsilon_i) + \|s_i\| \overline{J}_i(X, N)
\] (4.81)

Which results in
\[
\dot{\Lambda}_i \leq -\|s_i\| \epsilon_i
\] (4.82)
Thus, the reaching condition is met, sliding occurs in finite time and the control input described in Equation 4.64 for general dynamics described in Equation (4.59) is stable in the sense of Lyapunov.

4.6 Simulation Results and Discussion

4.6.1 Quadcopter Swarm

Using the quadcopter model detailed in Equation (2.1) for a homogeneous swarm, attitude Sliding Mode Control Equation (2.6), modified altitude input Equation (2.8), desired roll and pitch Equations (2.7), and artificial potential field multi-agent position controller given as

\[ u_i = -K_i \text{sat}(s_i) \] (4.83)

for agent \( i \) where \( u_i = [u_{i,x} \quad u_{i,y} \quad u_{i,z}]^T \) are the virtual position inputs, \( K_i = [K_x \quad K_y \quad K_z]^T \) are the APF-SMC discontinuous gains which is the same for each agent, the signum function is replaced with a saturation function to reduce chattering and operate element wise on the sliding manifold \( s_i = [s_{i,x} \quad s_{i,y} \quad s_{i,z}]^T \), and

\[
\begin{align*}
    s_i &= \hat{x}_i + \nabla \hat{x}_i V_i(X_i, N) \\
    V_i(X_i, N) &= \sum_{k=1}^{M} \left( k_a \sqrt{||\hat{x}_i - \nu_k||^2 + c_s^2} - k_s e^{-\frac{||\hat{x}_i - \nu_k||^2}{a}} \right) \\
    &+ \sum_{j=1}^{N} \left( k_r e^{-\frac{||\hat{x}_i - \hat{x}_j||^2}{\alpha}} - k_l \sqrt{||\hat{x}_i - \hat{x}_j||^2 + c_s^2} \right)
\end{align*}
\] (4.6)

where \( \hat{x}_i = [x_i \quad y_i \quad z_i]^T \) is the position of agent \( i \). Without loss of generality, gains used in Equation (4.6) for simulations were chosen as \( k_a = \frac{4}{M}, \quad d_e = 3, \quad c_s = 30, \quad c_k = 0.5, \quad a = r, \quad \delta_l = 0.2 \frac{M}{N}, \quad \text{and} \quad \delta_s = 0.2 \frac{N}{M} \), with gains \( k_r \) and \( r \) are determined through Lemma 4.3.2. The discontinuous gains are selected as \( K_i = [2 \quad 2 \quad 20]^T \).
Figure 4.4 shows a group of 4 quadcopters ($N = 4$) with initial conditions:

$$x_0 = \begin{bmatrix}
25 & 20 & 0 \\
15 & 20 & 0 \\
20 & 25 & 0 \\
20 & 15 & 0
\end{bmatrix}$$ (4.84)

and 4 stationary attractive nodes ($M = 4$) stationary at $\nu$:

$$\nu = \begin{bmatrix}
2 & 2 & 5 \\
-2 & 2 & 5 \\
-2 & -2 & 5 \\
2 & -2 & 5
\end{bmatrix}$$ (4.85)

It is observed that all agents flock towards the attractive nodes without collision and settle in the intended formation without assigning specific agents of specific nodes.

Figure 4.5 shows a group of 4 quadcopters ($N = 4$) with initial conditions:
Figure 4.5: Trajectories tracking a 2-node formation

\[
\hat{x}_0 = \begin{bmatrix}
5 & 0 & 0 \\
-5 & 0 & 0 \\
0 & 5 & 0 \\
0 & -5 & 0 \\
\end{bmatrix}
\]  

and 2 attractive nodes \((M = 2)\) with trajectories

\[
\dot{\nu}(t) = \begin{bmatrix}
\sin(0.5t) + 3 & 0.25t & 5 \\
\sin(0.5t) - 3 & 0.25t & 5 \\
\end{bmatrix}
\]  

It can be seen that 2 quadcopters successfully track the moving nodes on their trajectories while the additional agents follow the swarm centroid while maintaining separation.

Figures 4.6a through 4.7 show a swarm of 4 quadcopters \((N = 4)\) with initial conditions:

\[
x_0 = \begin{bmatrix}
5 & 0 & 0 \\
-5 & 0 & 0 \\
0 & 5 & 0 \\
0 & -5 & 0 \\
\end{bmatrix}
\]  

41
Figure 4.6: Quadcopter swarm trajectories before and after agent failure

and 2 attractive nodes \((M = 2)\) stationary at:

\[
\nu(t) = \begin{bmatrix}
3 & 0 & 5 \\
-3 & 0 & 5
\end{bmatrix}
\] (4.89)

The agent trajectories during initial swarm aggregation can be seen in Figure 4.6a. At \(t = 30\) seconds in the simulation, Agent 1 is artificially sent out of the formation in the negative \(z\) direction. The full trajectories can be seen in Figure 4.6b. For clarity, Figure 4.7 shows the \(x\), \(y\), and \(z\) coordinates versus time on separate plots for all agents. When Agent 1 exits the formation, it is observed that the remaining agents adjust themselves to re-aggregate the desired formation. That is, Agents 3 moves to the now vacant node while Agent 4 finds a new equilibrium in the swarm. Agent 2 remains stationary during the re-aggregation. This is achieved solely through the interactions from the artificial potential field, noting that no assignments for agents to nodes are made \textit{a priori}.

The Artificial Potential Field method of multi-agent coordination may also be used for waypoint navigation. In this application a waypoint serves as an attractive node. Once an agent comes
Figure 4.7: Agent coordinates aggregating a 2 node formation with single-agent failure

(a) Agent trajectory in space
(b) Agent positions in $x$, $y$, and $z$ directions versus time

Figure 4.8: Lawnmower path achieved through waypoint navigation
within a threshold distance of the current waypoint, the attractive node changes to the next desired location. A waypoint list

\[
w = \begin{bmatrix}
0 & 0 & 5 \\
0 & 10 & 5 \\
5 & 10 & 5 \\
5 & -10 & 5 \\
10 & -10 & 5 \\
10 & 10 & 5 \\
15 & 10 & 5 \\
15 & -10 & 5 \\
20 & -10 & 5 \\
20 & 10 & 5
\end{bmatrix}
\] (4.90)

contains the corners of a basic lawnmower path. The waypoint iterates when the agent is within 1 meter of the current target. For this simulation gains were selected as \( k_a = 10, \ d_e = 2, \ c_s = 100, \ c_k = 10, \ a = r, \ \delta_a = 0.2, \) and \( \delta_l = 0.2. \) Gains \( k_r \) and \( r \) are determined through Lemma 4.3.2. The discontinuous gains are selected as \( K_i = [2 \quad 2 \quad 20]^T. \) Figure 4.8 shows simulated quadcopter initially at \((0, 2, 0)\) moving towards each successive waypoint, following the lawnmower path.

4.6.2 Point-Mass Swarms

Swarm Aggregation

Due to large computational loads, simulating numerous quadcopters requires a significant amount of time. As such, having shown proof of concept with relatively small quadcopter swarms, to lessen the computational load, larger swarms are simulated with simple point-mass, double integrator dynamics \( \ddot{x}_i = u_i. \) Furthermore, while it is shown in the previous section that the APF-SMC multi-agent coordinator is capable of controlling agents in three dimensions, the following simulations will be restricted to horizontal movement and interaction through the APF to the \( x-y \) plane to improve interpretability of Figures. APF gains \( k_a = \frac{4}{M}, \ d_e = 3, \ c_s = 30, \ c_k = 0.5, \ a = r, \ \delta_s = \frac{1}{5e \ M}, \) and \( \delta_l = \frac{1}{5e \ N}. \) Here the natural number \( e \) is used to prevent any singularities for dif-
ferent swarm sizes or numbers of nodes change. Gains $k_r$ and $r$ are determined through Lemma 4.3.2. The discontinuous gains are selected as $K_i = [20 \ 20]^T$.

For the following simulations initial positions $(x_{i,0}, y_{i,0})$ are selected randomly for each agent such that $x_{i,0}$ and $y_{i,0} \in [15, 25]$. Four stationary nodes (M=4) are used at

$$
\nu = \begin{bmatrix}
2 & 2 \\
-2 & 2 \\
-2 & -2 \\
2 & -2 \\
\end{bmatrix}.
$$

(4.91)

Swarms of eight (N=8) and twenty (N=20) are considered and given sufficient time for all agents to come to rest. The resulting trajectories are shown in Figures 4.9a and 4.9b, respectively. It is observed that the 8-agent swarm does not adequately aggregate the desired formation when compared to the swarm of 20 agents. Aggregation of the 8-agent swarm is improved by changing the desired equilibrium distance to $d_e = 2$, as seen in Figure 4.10a. Further improvements can be made by increasing the steepness factor to $c_s = 3000$ as shown in Figure 4.10b. However, doing this in real world application may cause instability from demanding high accelerations in the system or causing actuator saturation.

**Equilibrium Distance**

In Chapter 4 it is shown that choosing APF gains through Lemma 4.3.2 that agents should settle at a specified equilibrium distance $d_e$ as defined in Lemma 4.3.1. Simulated swarms with eight agents (N=8) aggregate around a single attractive node (M=1) at coordinates (0,0). Figures 4.11a and 4.11b show the resulting agent trajectories with $d_e = 2$ and $d_e = 5$, respectively.

**Swarms with Limited Communication**

For all simulations above, the swarm uses global communication, that is, each agent has full knowledge of all other agents in the swarm and the APF interactions are calculated accordingly. In real applications, such a communication scheme can be overly complicated and computationally intense. As such, limited communication between agents is investigated through simulation.
Figure 4.9: Large swarms with formation and point mass dynamics

(a) 8 agents, 4 nodes

(b) 20 agents, 4 nodes

Figure 4.10: 8-agent swarms with 4-node formation and point mass dynamics

(a) Steepness gain $c_s = 30$

(b) Steepness gain $c_s = 3000$
Considering the swarms in Figures 4.9a and 4.9b where $d_e = 3$ and $c_s = 30$, communication is restricted such that each agent only has knowledge of agents within a radius of $1.5d_e$ (4.5 meters), effectively limiting communication to adjacent agents. This both greatly limits communication between agents and encloses the settling point at the equilibrium distance from each agent. The resulting swarm trajectories, shown in Figures 4.12a and 4.12b, exhibit behavior extremely similar to that with global communication indicating global communication is not necessary to maintain stability and aggregate a formation using this APF method of swarm control.

**Additional Swarm Logic**

While the initial intention of the proposed multi-agent coordinator was to aggregate a formation without assigning agents to specific positions, some configurations are not easily formed, such as an in-line formation shown in Figure 4.13a. However, by assigning specific nodes to interact only with specific agents through the APF, the formation is accurately aggregated, as shown in Figure 4.13b. Additional logic may be introduced to improve or expand swarm behavior. This can include, but is not limited to, node occupancy, a swarm point where supplemental agents gather away from the formation, and dynamic node assignment.
Figure 4.12: Large swarms with formation and point mass dynamics with limited communication

(a) 8 agents, 4 nodes  
(b) 20 agents, 4 nodes

Figure 4.13: Inline formation with and without assigning agents to specific nodes

(a) No node assignment  
(b) With node assignment
CHAPTER 5
EXPERIMENTAL DEVELOPMENT

The UNH QuadSat Swarm senior project team, with the guidance of graduate researchers from the UNH Advanced Controls Lab, designs and builds quadcopter platforms to aid in the experimental validation of graduate level research. The main components of the quadcopter are

- Teensy 3.5 Microcontroller
- Adafruit NXP Precision 9 DOF IMU
- Marvelmind Indoor Navigation System
- Turnigy Multistar 980 Kv 14-Pole Brushless DC Motor
- 30A Electronic Speed Controller (ESC)
- 8x4.5 Propeller
- Raspberry Pi 3
- Xbee Series 2 and USB Dongle

The Teensy 3.5 microcontroller, which is compatible with Arduino libraries and devices, executes the main control loop to stabilize the quadcopter. It receives attitude (pitch $\phi$, roll $\theta$, and yaw $\psi$) measurements from the Adafruit NXP Precision 9 DOF IMU and positional ($x$, $y$, and $z$) measurements from the Marvelmind Indoor Navigation System. These signals are used in the feedback loop, detailed in Chapter 2 and tuned in Chapter 3, to stabilize and control the quadcopter states. The Xbee wireless communication modules are used to communicate a quadcopter’s position to a ground station. On the ground station computer, a Python version of the APF-SMC multi-agent
coordinator, detailed in Chapter 4, computes the control command signals $u_x$, $u_y$, and $u_z$ for all quadcopters. This is sent back to the quadcopter agents and used in the attitude control loop. Due to the speed of response of the quadcopter states, coding requirements from the sensors on board, and hardware limitations, the Teensy is not able to handle wireless communication and maintain attitude control in the required loop time. Therefore, the Raspberry Pi is used for all wireless communication with signals passed to the Teensy when needed.

At this time, the UNH QuadSat Swarm Team has implemented the PSO tuned SMC controller from Chapter 3. An attitude stabilization using the PSO tuned SMC attitude controller is shown in Figure 5.1. The high frequency noise present in the pitch and roll responses is likely due to vibrations in the Quadcopter frame caused by the propellers. It can be seen that the controller maintains stability for a prolonged period with an error often less than 2°. For the purpose of the QuadSat Swarm Team, this is considered highly effective.

Due to the low position sampling rate of the Marvelmind Indoor Navigation System, a preliminary PID position controller has not been implemented. While the APF-SMC multi-agent coordinator and communication between the quadcopters and ground station has been implemented, without a basic, proof of concept position controller experimentally validated, swarm coordina-
tion experiments cannot take place. Current work is under way to increase the sample rate of the Marvelmind system as well as examining alternative methods of position determination.
CHAPTER 6
CONCLUSION AND FUTURE WORK

This research demonstrated that Particle Swarm Optimization is an efficient and effective method of gain tuning for a Quadcopter attitude Sliding Mode Control scheme. A modified, decentralized Artificial Potential Field swarm coordinator based on a prior leader-follower scheme was developed and found to be stable using Sliding Mode Control. A gain selection method was proposed and shown to yield a desired settling distance between agents in a swarm. It was also shown that an Artificial Potential Field Sliding Mode Control algorithms can be used to effectively aggregate a specified multi-node formation with supplemental agents in a larger swarm available to replace agents which malfunction and are force to leave the formation and follow waypoint navigation. Additional logic, in the form of node assignment, was introduced and improved formation aggregation was shown.

The Particle Swarm Optimized attitude controller was implemented on a custom built, experimental quadcopter and found to be very effective. While position control is not yet available for experimental testing due to current hardware limitations, a significant portion of the preliminary work necessary to implement the APF-SMC multi-agent coordinator has been completed with the help of the UNH QuadSat Swarm senior project team.

Numerous topics for future work maybe be pursued based on the research presented here. Studies into the tracking accuracy of the APF-SMC multi-agent coordinator are immediately obvious. To improve this swarm controller, the work on PSO tuning maybe applied to selecting gains for the APF. Additionally, stability criteria may be investigated for agents converging to attractive nodes and minimal communication limits. Another immediate possibility for future work is the introduction of additional swarm logic including node occupancy, a swarming point for supplemental agents, and dynamic node assignment.
Future work should include the experimental validation of the APF-SMC swarm coordinator. Current progress indicates that this may be achieved in the near future. Future work on experimental implementation can include improved communication schemes to remove the ground station from the control loop to produce a completely decentralized swarm.

Other topics of research being performed by members of the Advanced Control Lab provide additional possible developments for the APF-SMC architecture. Hybrid APF and A* path planning may be incorporated with the inclusion of obstacle avoidance. Additional work considers using Earth-based quadcopters to test satellite control methods. It may be possible to equate the APF to a gravitational potential model to simulate satellite orbits in an experimental environment.


APPENDIX

PARTICLE SWARM OPTIMIZATION FIGURES

This page was unintentionally left blank.
Figure A.1: Cost of each particle’s controller at each step using equation (3.3) as the cost function and coupled responses ($N = 10, k_f = 100$)

Figure A.2: Controller gains for each particle at each step using equation (3.3) as the cost function and coupled responses ($N = 10, k_f = 100$)
Figure A.3: Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and coupled responses ($N = 10, k_f = 100$)

Figure A.4: Cost of each particle’s controller at each step using equation (3.3) as the cost function and coupled responses ($N = 100, k_f = 10$)
Figure A.5: Controller gains for each particle at each step using equation (3.3) as the cost function and coupled responses ($N = 100, k_f = 10$)

Figure A.6: Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and coupled responses ($N = 100, k_f = 10$)
Figure A.7: Cost of each particle’s controller at each step using equation (3.3) as the cost function and coupled responses ($N = 30, k_f = 100$)

Figure A.8: Controller gains for each particle at each step using equation (3.3) as the cost function and coupled responses ($N = 30, k_f = 100$)
Figure A.9: Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and coupled responses ($N = 30, k_f = 100$)

Figure A.10: Cost of each particle’s controller at each step using equation (3.3) as the cost function and decoupled responses ($N = 10, k_f = 100$)
Figure A.11: Controller gains for each particle at each step using equation (3.3) as the cost function and decoupled responses ($N = 10$, $k_f = 100$)

Figure A.12: Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and decoupled responses ($N = 10$, $k_f = 100$)
Figure A.13: Cost of each particle’s controller at each step using equation (3.3) as the cost function and decoupled responses ($N = 100$, $k_f = 10$)

Figure A.14: Controller gains for each particle at each step using equation (3.3) as the cost function and decoupled responses ($N = 100$, $k_f = 10$)
Figure A.15: Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and decoupled responses ($N = 100, k_f = 10$)

Figure A.16: Cost of each particle’s controller at each step using equation (3.3) as the cost function and decoupled responses ($N = 30, k_f = 100$)
Figure A.17: Controller gains for each particle at each step using equation (3.3) as the cost function and decoupled responses ($N = 30, k_f = 100$)

Figure A.18: Quadcopter responses using the final optimized controller found using equation (3.3) as the cost function and decoupled responses ($N = 30, k_f = 100$)
Figure A.19: Cost of each particle’s controller at each step using equation (3.5) as the cost function and coupled responses ($N = 10, k_f = 100$)

Figure A.20: Controller gains for each particle at each step using equation (3.5) as the cost function and coupled responses ($N = 10, k_f = 100$)
Figure A.21: Quadcopter responses using the final optimized controller found using equation (3.5) as the cost function and coupled responses ($N = 10, k_f = 100$)

Figure A.22: Cost of each particle’s controller at each step using equation (3.5) as the cost function and coupled responses ($N = 100, k_f = 10$)
Figure A.23: Controller gains for each particle at each step using equation (3.5) as the cost function and coupled responses ($N = 100$, $k_f = 10$)

Figure A.24: Quadcopter responses using the final optimized controller found using equation (3.5) as the cost function and coupled responses ($N = 100$, $k_f = 10$)
Figure A.25: Cost of each particle’s controller at each step using equation (3.5) as the cost function and coupled responses ($N = 20, k_f = 100$)

Figure A.26: Controller gains for each particle at each step using equation (3.5) as the cost function and coupled responses ($N = 30, k_f = 100$)
Figure A.27: Quadcopter responses using the final optimized controller found using equation (3.5) as the cost function and coupled responses ($N = 30, k_f = 100$)