

A RIGID BODY FRAMEWORK FOR MODELING AND SIMULATION OF  
FORMATION OF AUTONOMOUS AGENTS

by  
SUGURU SATO

Presented to the Faculty of the Graduate School of  
The University of Texas at Arlington in Partial Fulfillment  
of the Requirements  
for the Degree of

MASTER OF SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2023

A RIGID BODY FRAMEWORK FOR MODELING AND SIMULATION OF  
FORMATION OF AUTONOMOUS AGENTS

The members of the Committee approve the Masters  
Thesis of Suguru Sato

Kamesh Subbarao  
Supervising Professor

---

Julio César Benavides

---

Shuo Linda Wang

---

---

---

Dean of the Graduate School

---

Copyright © by Suguru Sato 2023

All Rights Reserved

To my father Eiichi Sato and mother Mika Sato for believing in me and their financial support, and to my friends and peers who supported me, without their love and support, the completion of this work would not have been possible.

## Acknowledgements

I would like to express my deepest gratitude and thanks to my supervising professor Dr. Kamesh Subbarao for accepting me in the Aerospace Systems Laboratory (ASL) as well as providing me with a wonderful opportunity to work with him on this research project. His thought process and attitude towards problem solving as well as his significant amount of knowledge in a wide range of subjects always surprised and motivated me to face the problem and make progress day-by-day to achieve what I was out for and taught me that a strong foundation and humble attitude towards the challenge is the most important ingredient for a successful outcome. Without his guidance, patience, and persistent help as well as a number of discussions on this project, the completion of this research would not have been possible. It has been an honor for me to work with such an outstanding researcher and have him as my mentor during my master's degree.

I would also like to thank Dr. Julio César Benavides for recommending me as a Graduate Teaching Assistant (GTA) for MAE 3405 (Flight Dynamics). While the class is an undergraduate class, being a CA as well as his well constructed lecture gave me the opportunity to relearn important subjects related to my research and to learn how to develop organized simulations. Moreover, his genuine and respectful manner towards his students as well as his enthusiasm towards teaching has become one of the motivations to seek a Ph.D. degree in aerospace engineering and to become a professor as one of my career goals.

A sincere thanks to my father and mother for supporting me unconditionally. Being from a middle class, supporting a child studying in a university in the United

States would not have been easy financially as well as mentally. I would also like to thank my friends Mark Liu, Omar Khammash, Braulio Mora, and Elshaday Zeray for motivating me with their brilliant minds. Lastly, I would like to thank my girlfriend Naomi Celis Licon for being patient and encouraging me through challenging times.

November 09, 2023

Abstract

A RIGID BODY FRAMEWORK FOR MODELING AND SIMULATION OF  
FORMATION OF AUTONOMOUS AGENTS

Suguru Sato, M.S.

The University of Texas at Arlington, 2023

Supervising Professor: Kamesh Subbarao

Formation motion, wherein multiple agents operate within close proximity while maintaining precise relative positions and coordinated movements, has a significant impact in the realm of Uncrewed Aerial Systems (UAS). Formation establishment, and station-keeping have been critical research areas as several missions can be conceived and accomplished by groups of “simple” autonomous agents. Advances in this research have led to multiple advantages in areas such as reconnaissance, aerospace exploration, map building, search and rescue, etc.; particularly, if the formation motion is inspired by nature and natural phenomena such as flocks of birds, school of fish, smooth-streamline hydrodynamics, magnetic fields, and electrical charges. In such formulations, the formation motion is often captured by a few rules, which are collision avoidance, velocity matching, and flock centering. Besides the examples seen in nature, our research is motivated by the characteristics of a rigid body. By considering the autonomous agents as fixed nodes on a rigid body, the trajectory of each agent on the rigid body is automatically determined based on the behavior and motion of the rigid body itself. Therefore, there is no need to design their trajectories

individually. Moreover, since distances between agents are held constant, collision free trajectories are automatically determined. We call this a Virtual Rigid Body (VRB) formation as there is no physical rigid connection between agents, and the virtual rigidity between agents is accomplished via constraint forces.

The main contributions of this research are thorough mathematical modeling, control law development, and simulation of this VRB framework for formation motion. The thesis synthesizes the six-degree-of-freedom (6-DOF) equations of motion associated with the framework for an arbitrary number of autonomous agents. Reference motion is prescribed for the aggregated rigid body – velocity of the center of mass and the angular velocity. Using these and the rigid body structure specification, local trajectories are calculated to perform the formation establishment. The thesis also develops local control laws for each agent using linear quadratic control methods, designed to ensure formation establishment, and station-keeping. Extensive simulations within an artificial environment are performed to illustrate the effectiveness of the modeling framework. Several example situations are considered such as – formation establishment, reconfiguration, re-orientation, and station-keeping.



## Table of Contents

Acknowledgements . . . . .	v
Abstract . . . . .	vii
List of Illustrations . . . . .	xi
List of Tables . . . . .	xiv
Chapter	Page Chapter
1. Introduction . . . . .	1
1.1 Background Development . . . . .	1
1.2 Problem Statement and Motivation . . . . .	4
2. EQUATIONS OF MOTION OF A MULTIAGENT SYSTEM . . . . .	8
2.1 Translational Dynamics . . . . .	8
2.2 Rotational Dynamics . . . . .	10
2.3 Rotational Kinematics . . . . .	13
2.4 Translational Kinematics . . . . .	14
2.5 Equations of Motion Summary . . . . .	15
3. CONSTRAINT FORCE SYSNTHESIS FOR AN N-AGENT VIRTUAL RIGID BODY FORMATION . . . . .	16
3.1 General Constraint Force Development . . . . .	16
3.2 Constraint Force Stabilization . . . . .	21
4. CONSTRAINT FORCE VALIDATION VIA SIMULATIONS . . . . .	24
5. FORMATION BODY FRAME AND INPUT DECOUPLING . . . . .	31
5.1 Body Frame Attachment . . . . .	31
5.2 Input Decoupling . . . . .	33

6. SYSTEM SYNTHESIS AND SIMULATION . . . . .	35
6.1 Straight Line Trajectory . . . . .	36
6.2 Rolling Maneuver . . . . .	38
6.3 Pitching Maneuver . . . . .	39
6.4 Yawing Maneuver . . . . .	40
7. APPLICATION EXAMPLES . . . . .	43
7.1 Single Waypoint Simulations . . . . .	44
7.1.1 Linear Quadratic Regulator . . . . .	44
7.1.2 Local Voting Protocol . . . . .	49
7.1.3 Trajectory Tracking . . . . .	52
7.2 Multiple Waypoint Simulations . . . . .	57
8. FORMATION WITH HIGHER NUMBER OF AGENTS . . . . .	63
8.1 Formation Validation . . . . .	63
8.2 Body Frame Attachment . . . . .	65
8.3 Integrated System Validation . . . . .	66
8.4 Applicaiton Example . . . . .	68
9. SUMMARY, CONCLUSION, AND FUTURE WORK . . . . .	75
9.1 Summary . . . . .	75
9.2 Conclusion . . . . .	76
9.3 Future Work . . . . .	77
Appendix	
A. Additional Formation Cases . . . . .	79
References . . . . .	88
Biographical Statement . . . . .	92

## List of Illustrations

Figure	Page
1.1 (a) Flock of birds and (b) School of fish . . . . .	2
1.2 (a) Airborne refueling and (b) Space formation flying . . . . .	3
1.3 Wind tunnel test on a car . . . . .	5
2.1 Representative collection of particles used to model a multi-agent system	8
3.1 Illustration of the relations of $m_i$ and $m_j$ . . . . .	16
4.1 Simulink model for formation establishment . . . . .	24
4.2 Simulink model architecture for formation establishment . . . . .	25
4.3 Three-agent cases constraint graphs. (Left): Equilateral triangle forma- tion, (Right): Linear formation. Orange lines - applied constraints . .	26
4.4 Equilateral triangle formation establishment . . . . .	27
4.5 Three-agent linear formation . . . . .	28
4.6 Three-agent formation reconfiguration . . . . .	29
4.7 Three-agent triangle-to-linear formation under translational motion . .	30
5.1 Formation with body frame . . . . .	31
6.1 Synthesized Simulink model . . . . .	35
6.2 Synthesized Simulink model architecture . . . . .	36
6.3 Formation under the constant input - Linear motion . . . . .	37
6.4 Formation under the constant input - Rolling motion . . . . .	39
6.5 Formation under the constant input - Pitching motion . . . . .	40
6.6 Formation under the constant input - Yawing motion . . . . .	41
6.7 Results of Euler angles . . . . .	41

7.1	Environment setup . . . . .	43
7.2	Single waypoint simulation results via LQR . . . . .	47
7.3	Single way point simulation results via LQR: Control Inputs . . . . .	48
7.4	Single waypoint simulation results via Local Voting Protocol . . . . .	50
7.5	Single waypoint simulation result via Local Voting Protocol: Control Inputs . . . . .	51
7.6	Reference trajectory . . . . .	53
7.7	Reference position and velocity data . . . . .	54
7.8	Single waypoint simulation results via trajectory tracking . . . . .	55
7.9	Single way-point simulation results for trajectory tracking: control in- puts of each agent . . . . .	56
7.10	Environment descriptions and mission details . . . . .	57
7.11	Simulink Architecture for Multi-Waypoint flight . . . . .	59
7.12	Multiple waypoint simulation results via LQR: Formation trajectory .	59
7.13	Multiple waypoint simulation results via LQR: Distances between agents	60
7.14	Multiple waypoint simulation results via LQR: Formation Euler angles (left), Formation Angular Rates (right) . . . . .	61
7.15	Multiple waypoint simulation results via LQR: Agents' Control Inputs	62
8.1	Constraint graph of cubic formation . . . . .	63
8.2	Cubic formation simulation results . . . . .	65
8.3	Eight-agent formation, body frame orientation . . . . .	65
8.4	Cubic formation single way point simulation results . . . . .	67
8.5	Cubic formation single way point simulation: Control inputs on agents	68
8.6	Environment and mission specifications . . . . .	69
8.7	Eight-agent VRB formation simulation result . . . . .	71
8.8	Eight-agent VRB formation simulation result: Distances between agents	72

8.9	Eight-agent VRB formation simulation results: Euler angles and angular rates . . . . .	73
8.10	Eight-agent VRB formation simulation results: Control Input . . . . .	73
A.1	Two agent formation result (Left): agent trajectories, (Right): Time history of distance between the two agents. . . . .	80
A.2	Four-agent cases constraint graph. (Left): Equilateral tripod formation, (Middle): Square formation, (Right): Linear formation. Orange lines - applied constraints. . . . .	80
A.3	Four-agent equilateral tripod formation simulation results . . . . .	81
A.4	Four-agent square formation simulation results . . . . .	81
A.5	Four-agent linear formation simulation results . . . . .	82
A.6	Five-agent case constraint graph. (Left): Equilateral pyramid formation, (Middle): Pentagon formation, (Right): Linear formation. Orange lines - applied constraints. . . . .	82
A.7	Equilateral pyramid formation simulation results . . . . .	83
A.8	Pentagon formation simulation results . . . . .	83
A.9	Linear formation simulation results . . . . .	84
A.10	Six-agent case constraint graph. (Left): Equilateral Dual-pyramid formation, (Middle): Hexagon formation, (Right): Linear formation. Orange lines - applied constraints. . . . .	84
A.11	Equilateral dual-pyramid formation simulation results . . . . .	85
A.12	Hexagon formation simulation results . . . . .	85
A.13	Linear formation simulation results . . . . .	86
A.14	Octagon formation simulation results . . . . .	86
A.15	Linear formation simulation results . . . . .	87

## List of Tables

Table	Page
4.1 Formation establishment input specifications . . . . .	25
6.1 Formation linear motion input specifications . . . . .	37
6.2 Formation roll motion input specifications . . . . .	38
6.3 Formation pitch motion input specifications . . . . .	39
6.4 Formation yaw motion input specifications . . . . .	40
7.1 Simulation specifications for single waypoint case . . . . .	44
7.2 Linear Quadratic Regulator Components . . . . .	46
7.3 Local Voting Protocol Components . . . . .	49
7.4 Equations used for optimal trajectory generation . . . . .	53
7.5 Simulation specifications for multiple-waypoint case . . . . .	58
8.1 Eight-agent case formation establishment input specifications . . . . .	64
8.2 Waypoint specificaitons for single waypoint case . . . . .	66
8.3 Simulation specifications for multiple-waypoint case: Eight-agent case .	70

## Chapter 1

### Introduction

#### 1.1 Background Development

Formation flight, wherein multiple aerial vehicles operate in close proximity while maintaining precise relative positions and coordinated movements, represents a significant impact in the realm of uncrewed systems. It also has been identified as an enabling technology for many of the NASA's 21st century missions [1]. In addition to aerial vehicles, such a system has also been applied towards a wide range of applications, such as mechanical systems, ships, aircraft, uncrewed vehicles, spacecraft, etc. [2]. Formations can also be often observed in the biological systems. In migrating bird flocks, as shown in figure 1.1(a), every bird operates within an upwash region created by the collective presence of all other birds ahead in the formation, resulting in a decrease for required power for individuals in the entire formation [3]. In a school of fish, figure 1.1(b), not only do the fish save their energy, but they also make themselves look bigger to avoid predation.



(a)



(b)

Figure 1.1. (a) Flock of birds and (b) School of fish<sup>1</sup>.

The problems of formation system and control are critical research areas in the aerospace engineering as some missions require groups of uncrewed aerial vehicles (UAVs), leading to multiple advantages in areas such as radio signal detection, reconnaissance, exploration, map building, and search and rescue [4]. For example, in the defense field, as shown in figure 1.2(a), there can be used for airborne refueling as well as quick deployment of troops and vehicles [5]. By employing formation flight, surveillance and reconnaissance on larger area can be achieved while improving the efficiency of the entire fleet. In the medical field, uncrewed aerial vehicles can be used for delivering health supplies in emergencies; moreover, by using multiple UAVs in a formation, more supplies can be delivered faster in such situations [6]. Similarly, as shown in figure 1.2(b), spacecraft formation has also been proposed to facilitate distributed sensing tasks such as gravity field mapping, atmospheric data sampling, co-observations, etc. [7].

---

<sup>1</sup>(a) National Geographic, url:<https://www.nationalgeographic.com/animals/article/these-birds-flock-in-mesmerizing-swarms-why-is-still-a-mystery>

(b) A. Liemena, “School of Fish Swimming Underwater”, url: <https://www.pexels.com/photo/school-of-fish-swimming-underwater-7826494/>





(a)



(b)

Figure 1.2. (a) Airborne refueling and (b) Space formation flying <sup>2</sup>.

While formation flight can be developed manually, it is usually performed autonomously by using corresponding algorithms and concepts to reduce human-errors. As approaches to achieve formations, there are roughly three methods such as leader-following, behavioral methods, and virtual structures [8]. In leader-follower formation, the leader is commanded to fly some predefined trajectories while followers are controlled to maintain relative position with respect to the leader [9]. The behavioral approach prescribes a set of behaviors including trajectory tracking, collision and obstacle avoidance, etc., and the entire formation is treated as a single body in the virtual structure approach [8].

Our research is motivated by the characteristics of rigid bodies and can be categorized as virtual structure approach. Ref. [10] achieved the virtual structure formation by planning a trajectory for each vehicle. While this is one of the meth-

---

<sup>2</sup>(a) J. Simms, “Refueling the SR-71”, *Aerotechnews*, url:<https://www.aerotechnews.com/blog/2020/08/12/refueling-the-sr-71/>

(b) NASA, “NASA’s Starling Cubesats To Demo Autonomous Formation Flying”, *Aviation Week*, url:<https://aviationweek.com/defense-space/space/nasas-starling-cubesats-demo-autonomous-formation-flying>

ods, we impose the rigidity of the formation in a way to treat the formation as if it was a rigid body, which results in no need for trajectory planning for each vehicle. For the formation establishment and maintenance, the concept of constraint force is employed similar to [5, 7, 11–13]. When we consider a rigid body as a group of particles, the constraint force is an internal force acting between the particles to constrain the relative motion of a particle to another. The constraint force is developed by using d’Alembert’s principle of virtual work, which states that constraint forces should not do any external work on the system [14]. To achieve constraint stabilization, a proportional-integral-derivative (PID)-like control law based on Baumgarte stabilization method is employed similar to [7, 8, 12].

While the concept of constraint force has been addressed in the literature, few papers address a virtual structure formation as if it were an independent rigid body by synthesizing its governing 6-DOF equations of motion. This leads us to the motivation and problem statement of this research, which is discussed in the following section.

## 1.2 Problem Statement and Motivation

Along with the recent surge of development in autonomous agents, researches in formation motion has also been gaining more attention as it allows a group of agents to efficiently fulfill missions that an individual agent cannot [15]. Formation motion is often captured by using a few rules such as collision avoidance, velocity matching, and flock centering [15]. Formation motion, as well as cooperative motion of autonomous agents that satisfy these rules, are often motivated by nature and natural phenomena. As it was mentioned in the previous section, migrating birds arrange specific formations so that the entire group can save energy and fly longer distances. They follow the formation rules through visual signals and from the shock waves propagating from neighbors [15]. Large flocks of birds as well as schools of fish

also achieve cooperative motion that satisfy those rules. As it is known from fluid dynamics, streamlines of laminar flow streamlines do not collide with each other; thus, collision-free trajectories of each agent may be constructed if one adopts a hydrodynamics approach to formations [16,17]. Known as the potential field approach, attraction and repulsion characteristics of magnetic poles or electrical charges can also be assigned for cooperative flight formations to satisfy those rules [18,19].

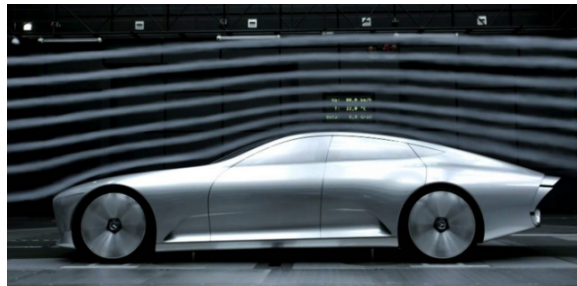


Figure 1.3. Wind tunnel test on a car <sup>3</sup>.

While collective motions in biological systems and natural phenomena often become motivation for the development of cooperative or formation flight modeling and simulation, our motivation in this research is based on the characteristics of rigid bodies. A rigid body holds several advantages when it comes to formation motion. As it is easily imagined, the trajectory of each point fixed on a rigid body is automatically determined based on the behavior of the rigid body itself; thus there is no need to individually design their trajectories as long as desired behavior of the rigid body is commanded. Moreover, the trajectories are collision-free since distances between points are held constant because of rigidity. We call this formation as *Virtual Rigid Body (VRB)* formation since an established formation works as a rigid body while

---

<sup>3</sup>“Turbulence - What a Drag It Is When You Drive”, *engineering.com*, url:<https://www.engineering.com/story/turbulence---what-a-drag-it-is-when-you-drive>

there is no physical connection between agents. The main contribution of this research is the thorough development of mathematical modeling and simulation of distance-based virtual structure formation including its formation establishment, formation maintenance, reconfiguration, re-orientation, and station-keeping.

This research is roughly broken down into three steps: development of the equations of motion for a multiagent system, development of the constraint forces, and model verification via simulations. The rest of the thesis is organized as follows. Chapter 2 develops 6-DOF governing equations of motion of a multiagent system without the assumption of rigidity. This means that terms that are usually eliminated from the rigid body assumption appear in the equations since we also aim for formation reconfiguration, and there is no physical rigidity in the formation. Chapter 3 addresses the VRB formation wherein desired distances between agents are achieved and stabilized. This is done by employing the concept of the constraint force. The constraint forces are synthesized by utilizing d'Alembert's principle of virtual work, constraint sensitivities (Lagrange multipliers) and constraint stabilization using the Baumgarte stabilization technique. Chapter 4 addresses the first phase of the model validations. In Chapter 4, formation establishments using the constraint forces are confirmed via simulations with multiple formation configurations including formation reconfiguration. In Chapter 5, necessary components for the formation are developed so that the 6-DOF governing equations of motion works in the simulations. It addresses attachment of the body frame to the VRB formation as well as input decoupling. Components developed in the previous chapters such as the constraint forces, governing equations of motion, body frame attachment, and input decoupling, are integrated in a common framework. Starting from simple maneuvers such as linear motion, rolling, pitching, and yawing maneuvers, multiple simple situations are simulated in Chapter 6. In Chapter 7, multiple methods for calculating control

inputs, such as Linear Quadratic Regulator (LQR), Local voting Protocol [15], and trajectory tracking algorithm utilizing open-loop Lyapunov control are integrated in the framework. All methods are validated for waypoint tracking. After integration of the control inputs, the model is simulated in a multiple waypoint environment. Finally, more extensive simulations are performed with an eight-agent case to show that the framework is scalable to handle complex mission scenarios.

## Chapter 2

### EQUATIONS OF MOTION OF A MULTIAGENT SYSTEM

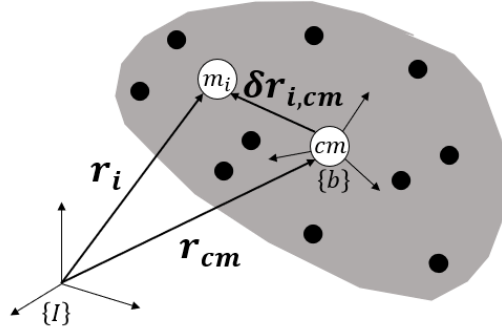


Figure 2.1. Representative collection of particles used to model a multi-agent system.

#### 2.1 Translational Dynamics

Linear momentum of  $m_i$  is expressed as follows.

$$\mathbf{p}_i = \mathbf{M}_i \mathbf{v}_i \quad (2.1)$$

Where, by letting  $d$  be the dimension of the system,  $\mathbf{M}_i = m_i \mathbf{I}_{d \times d}$  and  $\mathbf{p}_i$ ,  $\mathbf{v}_i \in \mathbb{R}^{d \times 1}$ . By representing a relative vector quantity  $\mathbf{a}$  of  $\alpha$  with respect to  $\beta$  as  $\delta \mathbf{a}_{\alpha, \beta}$ , the absolute velocity of  $m_i$  is expressed as  $\mathbf{v}_i = \mathbf{v}_{cm} + \delta \mathbf{v}_{i, cm}$ , where  $\mathbf{v}_{cm}$  is the absolute velocity of the system center of mass. Thus, equation (2.1) can be rewritten as follows.

$$\mathbf{p}_i = \mathbf{M}_i (\mathbf{v}_{cm} + \delta \mathbf{v}_{i, cm}) \quad (2.2)$$

By considering  $\delta\mathbf{v}_{i,cm}$  as described in the body frame associated with the multi-agent system, which is expressed as  $(\cdot)_B$ , equation (2.2) is further expressed as follows by using the transport theorem.

$$\mathbf{p}_i = \mathbf{M}_i \left\{ \mathbf{v}_{cm} + (\delta\mathbf{v}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta\mathbf{r}_{i,cm})_B \right\} \quad (2.3)$$

Where  $\boldsymbol{\omega}_B$  is the angular velocity of the multiagent system about its body axes. By taking derivative of equation (2.3) with respect to time, the force acting on  $\mathbf{M}_i$ ,  $\mathbf{f}_i$ , is calculated as follows.

$$\frac{I d}{dt} \mathbf{p}_i = \mathbf{f}_i = \mathbf{M}_i \left[ \frac{I d}{dt} \mathbf{v}_{cm} + \frac{I d}{dt} (\delta\mathbf{v}_{i,cm})_B + \frac{I d}{dt} \left\{ \boldsymbol{\omega}_B \times (\delta\mathbf{r}_{i,cm})_B \right\} \right] \quad (2.4)$$

By utilizing the transport theorem, equation (2.4) becomes,

$$\begin{aligned} \mathbf{f}_i &= \mathbf{M}_i \dot{\mathbf{v}}_{cm} + \mathbf{M}_i \left[ \frac{B d}{dt} (\delta\mathbf{v}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta\mathbf{v}_{i,cm})_B + \left\{ \frac{I d}{dt} \boldsymbol{\omega}_B \right\} \times (\delta\mathbf{r}_{i,cm})_B \right. \\ &\quad \left. + \boldsymbol{\omega}_B \times \left\{ \frac{I d}{dt} (\delta\mathbf{r}_{i,cm})_B \right\} \right] \\ &= \mathbf{M}_i \dot{\mathbf{v}}_{cm} + \mathbf{M}_i \left[ (\delta\dot{\mathbf{v}}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta\mathbf{v}_{i,cm})_B + \dot{\boldsymbol{\omega}}_B \times (\delta\mathbf{r}_{i,cm})_B \right. \\ &\quad \left. + \boldsymbol{\omega}_B \times \left\{ \frac{B d}{dt} (\delta\mathbf{r}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta\mathbf{r}_{i,cm})_B \right\} \right] \\ &= \mathbf{M}_i \dot{\mathbf{v}}_{cm} + \mathbf{M}_i \left[ (\delta\dot{\mathbf{v}}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta\mathbf{v}_{i,cm})_B + \dot{\boldsymbol{\omega}}_B \times (\delta\mathbf{r}_{i,cm})_B \right. \\ &\quad \left. + \boldsymbol{\omega}_B \times (\delta\mathbf{v}_{i,cm})_B + \boldsymbol{\omega}_B \times \left\{ \boldsymbol{\omega}_B \times (\delta\mathbf{r}_{i,cm})_B \right\} \right] \end{aligned} \quad (2.5)$$

By simplifying equation (2.5), following equation is obtained.

$$\begin{aligned} \mathbf{f}_i &= \mathbf{M}_i \dot{\mathbf{v}}_{cm} + \mathbf{M}_i \left[ (\delta\dot{\mathbf{v}}_{i,cm})_B + 2 \boldsymbol{\omega}_B \times (\delta\mathbf{v}_{i,cm})_B + \dot{\boldsymbol{\omega}}_B \times (\delta\mathbf{r}_{i,cm})_B \right. \\ &\quad \left. + \boldsymbol{\omega}_B \times \left\{ \boldsymbol{\omega}_B \times (\delta\mathbf{r}_{i,cm})_B \right\} \right] \end{aligned} \quad (2.6)$$

Now, by solving equation (2.6) for  $(\delta\dot{\mathbf{v}}_{i,cm})_B$ , translational dynamics of a multiagent system is expressed as follows.

$$\begin{aligned}
(\delta\dot{\mathbf{v}}_{i,cm})_B = & -2 \boldsymbol{\omega}_B \times (\delta\mathbf{v}_{i,cm})_B - \dot{\boldsymbol{\omega}}_B \times (\delta\mathbf{r}_{i,cm})_B - \boldsymbol{\omega}_B \times \left\{ \boldsymbol{\omega}_B \times (\delta\mathbf{r}_{i,cm})_B \right\} \\
& - [\mathbf{M}_i]^{-1} \left\{ [\mathbf{T}]_f^B \mathbf{f}_{cm} - [\mathbf{T}]_f^B \mathbf{f}_i \right\}
\end{aligned} \tag{2.7}$$

Where  $[\mathbf{T}]_f^B$  represents a transformation matrix that transforms a frame in which a force is originally expressed to the body frame associated with the rigid body formation, and  $\mathbf{f}_{cm}$  is the input force acting on the system center of mass.

## 2.2 Rotational Dynamics

The angular momentum  $\mathbf{h}_{cm}$  about the system center of mass is described as follows.

$$\mathbf{h}_{cm} = \sum_{i=1}^N \delta\mathbf{r}_{i,cm} \times \mathbf{M}_i \delta\dot{\mathbf{r}}_{i,cm} \tag{2.8}$$

By considering that the relative position of  $m_i$  with respect to the center of mass (described in the body frame of the multiagent system), equation (2.8) can be described as follows by using the transport theorem.

$$\mathbf{h}_{cm} = \sum_{i=1}^N (\delta\mathbf{r}_{i,cm})_B \times \mathbf{M}_i \left\{ (\delta\dot{\mathbf{r}}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta\mathbf{r}_{i,cm})_B \right\} \tag{2.9}$$



Equation (2.9), can be expanded and modified as follows.

$$\begin{aligned}
\mathbf{h}_{cm} &= \left\{ \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B \right\} + \left\{ \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i \boldsymbol{\omega}_B \times (\delta \mathbf{r}_{i,cm})_B \right\} \\
&= \left\{ \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B \right\} - \left\{ \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i \boldsymbol{\omega}_B \right\} \quad (2.10) \\
&= \left\{ \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B \right\} + \left\{ \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B^{\times} (\delta \mathbf{r}_{i,cm})_B^{\times T} \mathbf{M}_i \boldsymbol{\omega}_B \right\}
\end{aligned}$$

Where  $(\delta \mathbf{r}_{i,cm})_B^{\times}$  represents the skew symmetric matrix of the relative position,  $(\delta \mathbf{r}_{i,cm})_B$ . Since  $\sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B^{\times} (\delta \mathbf{r}_{i,cm})_B^{\times T} \mathbf{M}_i = [\mathbf{I}_{cm}]_B$ , equation (2.10) can be rewritten as follows.

$$\mathbf{h}_{cm} = \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B + [\mathbf{I}_{cm}]_B \boldsymbol{\omega}_B \quad (2.11)$$

Please note that if the system is a rigid body, the relative velocity,  $(\delta \dot{\mathbf{r}}_{i,cm})_B$ , is eliminated, and the familiar equation of angular momentum,  $\mathbf{h}_{cm} = [\mathbf{I}_{cm}]_B \boldsymbol{\omega}_B$ , appears. By taking a derivative of equation (2.11) with respect to time, torque acting on the center of mass of the multiagent system is obtained as follows.

$$\begin{aligned}
\frac{I d}{dt} (\mathbf{h}_{cm})_B &= \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times (\mathbf{f}_i)_B \\
&= \sum_{i=1}^N \left\{ \frac{I d}{dt} (\delta \mathbf{r}_{i,cm})_B \right\} \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B + \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i \left\{ \frac{I d}{dt} (\delta \dot{\mathbf{r}}_{i,cm})_B \right\} \\
&\quad + \left\{ \frac{I d}{dt} [\mathbf{I}_{cm}]_B \right\} \boldsymbol{\omega}_B + [\mathbf{I}_{cm}]_B \left\{ \frac{I d}{dt} \boldsymbol{\omega}_B \right\} \quad (2.12)
\end{aligned}$$

By applying the transport theorem, equation (2.12) can be rewritten as follows.

$$\begin{aligned}
\sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times (\mathbf{f}_i)_B &= \sum_{i=1}^N \left\{ (\delta \dot{\mathbf{r}}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta \mathbf{r}_{i,cm}) \right\} \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B \\
&+ \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i \left\{ (\delta \ddot{\mathbf{r}}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta \dot{\mathbf{r}}_{i,cm})_B \right\} \\
&+ \left\{ [\dot{\mathbf{I}}_{cm}]_B + \boldsymbol{\omega}_B \times [\mathbf{I}_{cm}]_B \right\} \boldsymbol{\omega}_B + [\mathbf{I}_{cm}] \dot{\boldsymbol{\omega}}
\end{aligned} \tag{2.13}$$

Since  $(\delta \dot{\mathbf{r}}_{i,cm})_B \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B = 0$ , equation (2.13) can be simplified as follows:

$$\begin{aligned}
\sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times (\mathbf{f}_i)_B &= \boldsymbol{\omega}_B \times \sum_{i=1}^N (\delta \mathbf{r}_{i,cm}) \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B \\
&+ \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i (\delta \ddot{\mathbf{r}}_{i,cm}) \\
&+ \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i \boldsymbol{\omega}_B \times (\delta \dot{\mathbf{r}}_{i,cm})_B \\
&+ [\dot{\mathbf{I}}_{cm}]_B \boldsymbol{\omega}_B + \boldsymbol{\omega}_B \times [\mathbf{I}_{cm}]_B \boldsymbol{\omega} + [\mathbf{I}_{cm}]_B \dot{\boldsymbol{\omega}}
\end{aligned} \tag{2.14}$$

By solving equation (2.14) by  $\dot{\boldsymbol{\omega}}_B$ , we can finalize the equations of rotational dynamics as follows.

$$\begin{aligned}
\dot{\boldsymbol{\omega}}_B &= -[\mathbf{I}_{cm}]_B^{-1} \left[ \boldsymbol{\omega}_B \times \sum_{i=1}^N (\delta \mathbf{r}_{i,cm}) \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B + \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i (\delta \ddot{\mathbf{r}}_{i,cm}) \right. \\
&\quad + \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i \boldsymbol{\omega}_B \times (\delta \dot{\mathbf{r}}_{i,cm})_B + [\dot{\mathbf{I}}_{cm}]_B \boldsymbol{\omega}_B + \boldsymbol{\omega}_B \times [\mathbf{I}_{cm}]_B \boldsymbol{\omega}_B \\
&\quad \left. - \sum_{i=1}^N (\boldsymbol{\tau}_{i,cm})_B \right]
\end{aligned} \tag{2.15}$$

Where  $(\boldsymbol{\tau}_{i,cm})_B$  is the torque acting on the system center of mass.

### 2.3 Rotational Kinematics

The Euler kinematic equations in a 3-2-1 sequence of elementary rotation with angles  $\psi$ ,  $\theta$ , and  $\phi$  are expressed as follows.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_1(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_1(\phi)\mathbf{R}_2(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.16)$$

After performing this calculation, we can obtain the following expression for  $\dot{\phi}$ ,  $\dot{\theta}$ , and  $\dot{\psi}$ .

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\sin(\theta) \end{bmatrix}^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.17)$$

Which results in the following equations.

$$\begin{aligned} \dot{\phi} &= p + \tan(\theta) \left\{ q \sin(\phi) + r \cos(\phi) \right\} \\ \dot{\theta} &= q \cos(\phi) - r \sin(\phi) \\ \dot{\psi} &= \left\{ q \sin(\phi) + r \cos(\phi) \right\} / \cos(\theta) \end{aligned} \quad (2.18)$$

As equation (2.18) shows, there could be singularity due to  $\theta = \pi/2$ . To address kinematic singularities resulting from the description using Euler angles and ensure all-attitude capability [20], Quaternions are utilized for the rotational kinematics. They are currently employed in a variety of applications, including robotics,

calculation for guidance and navigation, attitude control, and animation [20]. The Quaternion kinematic equation is expressed as follows.

$$\begin{bmatrix} \dot{k}_0 \\ \dot{k}_1 \\ \dot{k}_2 \\ \dot{k}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix} \quad (2.19)$$

Where,  $\mathbf{k} = [k_0, k_1, k_2, k_3]^T$  is the attitude quaternion, and  $p$ ,  $q$ , and  $r$  represent the components of  $\boldsymbol{\omega}_B$ . Please note that after integration of equation (2.19), the quaternions are normalized to ensure  $\|\mathbf{k}\| = 1$ . By using the quaternions, the directional cosine matrix can be constructed as follows.

$$[\mathbf{C}] = \begin{bmatrix} (k_0^2 + k_1^2 - k_2^2 - k_3^2) & 2(k_1k_2 + k_0k_3) & 2(k_1k_3 - k_0k_2) \\ 2(k_1k_2 - k_0k_3) & (k_0^2 - k_1^2 + k_2^2 - k_3^2) & 2(k_2k_3 + k_0k_1) \\ 2(k_1k_3 + k_0k_2) & 2(k_2k_3 - k_0k_1) & (k_0^2 - k_1^2 - k_2^2 + k_3^2) \end{bmatrix} \quad (2.20)$$

## 2.4 Translational Kinematics

Since we consider that the properties of the center of mass of the multiagent system are described in the inertial frame, and the relative properties of  $m_i$  with respect to the center of mass are described in the body frame of the system, the relation,  $\mathbf{v}_i = \mathbf{v}_{cm} + \delta\mathbf{v}_{i,cm}$ , is described as follows.

$$(\mathbf{v}_i)_I = (\mathbf{v}_{cm})_I + [\mathbf{T}]_B^I \left\{ (\delta\mathbf{v}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta\mathbf{r}_{i,cm})_B \right\} \quad (2.21)$$

Where  $[\mathbf{T}]_B^I$  represents a transformation matrix from the multiagent system body frame to the inertial frame.

## 2.5 Equations of Motion Summary

A summary of the multiparticle system equations of motion is shown below.

### Translational Dynamics

$$\begin{aligned}
 (\delta \mathbf{v}_{i,cm})_B &= -2 \boldsymbol{\omega}_B \times (\delta \mathbf{v}_{i,cm})_B - \dot{\boldsymbol{\omega}}_B \times (\delta \mathbf{r}_{i,cm})_B - \boldsymbol{\omega}_B \times \left\{ \boldsymbol{\omega}_B \times (\delta \mathbf{r}_{i,cm})_B \right\} \\
 &\quad - [\mathbf{M}_i]^{-1} \left\{ [\mathbf{M}]_f^B \mathbf{f}_{cm} - [\mathbf{M}]_f^B \mathbf{f}_i \right\}
 \end{aligned} \tag{2.22}$$

### Rotational Dynamics

$$\begin{aligned}
 \dot{\boldsymbol{\omega}}_B &= -[\mathbf{I}_{cm}]_B^{-1} \left[ \boldsymbol{\omega}_B \times \sum_{i=1}^N (\delta \mathbf{r}_{i,cm}) \times \mathbf{M}_i (\delta \dot{\mathbf{r}}_{i,cm})_B + \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i (\delta \ddot{\mathbf{r}}_{i,cm}) \right. \\
 &\quad \left. + \sum_{i=1}^N (\delta \mathbf{r}_{i,cm})_B \times \mathbf{M}_i \boldsymbol{\omega}_B \times (\delta \dot{\mathbf{r}}_{i,cm})_B + [\dot{\mathbf{I}}_{cm}]_B \boldsymbol{\omega}_B + \boldsymbol{\omega}_B \times [\mathbf{I}_{cm}]_B \boldsymbol{\omega}_B - \sum_{i=1}^N (\boldsymbol{\tau}_{i,cm})_B \right]
 \end{aligned} \tag{2.23}$$

### Rotational Kinematics

$$\begin{bmatrix} \dot{k}_0 \\ \dot{k}_1 \\ \dot{k}_2 \\ \dot{k}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix} \tag{2.24}$$

### Translational Kinematics

$$(\mathbf{v}_i)_I = (\mathbf{v}_{cm})_I + [\mathbf{T}]_B^I \left\{ (\delta \mathbf{v}_{i,cm})_B + \boldsymbol{\omega}_B \times (\delta \mathbf{r}_{i,cm})_B \right\} \tag{2.25}$$

## Chapter 3

### CONSTRAINT FORCE SYNTHESIS FOR AN N-AGENT VIRTUAL RIGID BODY FORMATION

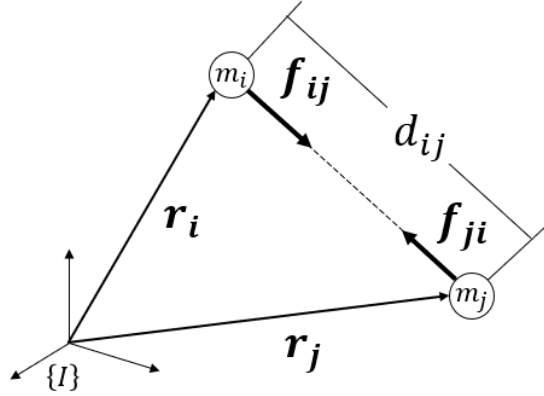


Figure 3.1. Illustration of the relations of  $m_i$  and  $m_j$ .

#### 3.1 General Constraint Force Development

A rigid body can be considered as a group of  $N$  particles that strictly maintain relative properties between one another. As [14] states, the force acting on the  $i$  th particle in a rigid body can be broken down into two subsets of forces as follows,

$$\mathbf{f}_i = \mathbf{f}_{i,E} + \sum_{j=1, j \neq i}^{N-1} \mathbf{f}_{ij} \quad (3.1)$$

where

$$\mathbf{f}_{i,E} = \mathbf{f}_{i,ext} + \mathbf{f}_{i,u} \quad (3.2)$$

$\mathbf{f}_{i,ext}$  is the vector sum of all external forces acting on mass  $m_i$ , such as aerodynamics forces, gravitational forces, and so on.  $\mathbf{f}_{i,u}$  is the vector sum of control inputs applied to each mass to achieve the desired trajectory of the formation center of mass. The inter-agent force,  $\mathbf{f}_{ij}$ , is the internal force acting on  $m_i$  due to the influence of  $m_j$  [14]. Within a rigid body, this force,  $\mathbf{f}_{ij}$ , is eliminated due to Newton's third law. The idea of a virtual rigid body formation uses this inter-agent force to constrain the motion of one agent relative to others. Since this force "constrains" the degree of rigidity of a multiparticle system, it is called a *constraint force*. By letting the sum of all constraint forces acting on  $m_i$  be  $\mathbf{f}_{i,C}$ , (i.e.  $\mathbf{f}_{i,C} = \sum_{j=1, j \neq i}^N \mathbf{f}_{ij}$ ), total force acting on  $m_i$  is expressed as follows.

$$\mathbf{f}_i = \mathbf{f}_{i,ext} + \mathbf{f}_{i,u} + \mathbf{f}_{i,C} \quad (3.3)$$

The constraint force can also be regarded as the control input for the VRB, in a sense since it is applied to each point mass to control the relative position of them to others to maintain the formation. However, let us consider  $\mathbf{f}_{i,u}$  to be the control inputs calculated to achieve the global mission such as way point flight, trajectory tracking, and so on, and the constraint force,  $\mathbf{f}_{i,C}$ , to be the control inputs that control the local behavior of each agent. Therefore, the equation of translational dynamics, equation (2.22), is further described as follows [11].

$$\begin{aligned} (\delta \dot{\mathbf{v}}_{i,cm})_B = & -2 \boldsymbol{\omega}_B \times (\delta \mathbf{v}_{i,cm})_B - \dot{\boldsymbol{\omega}}_B \times (\delta \mathbf{r}_{i,cm})_B - \boldsymbol{\omega}_B \times \left\{ \boldsymbol{\omega}_B \times (\delta \mathbf{r}_{i,cm})_B \right\} \\ & - \mathbf{M}_i^{-1} [\mathbf{T}]_f^B \left\{ \mathbf{f}_{cm} - (\mathbf{f}_{i,ext} + \mathbf{f}_{i,u} + \mathbf{f}_{i,C}) \right\} \end{aligned} \quad (3.4)$$

The purpose of this section is to develop an expression for the constraint force that establishes a formation and maintains it similar to [11]. The forces acting on point masses in the VRB can be collectively expressed as follows.

$$\mathbf{M}\ddot{\mathbf{r}} = \mathbf{f} = \mathbf{f}_E + \mathbf{f}_C \quad (3.5)$$

Where,  $\mathbf{M} = \begin{bmatrix} [\mathbf{M}_1] & 0_{3 \times 3} & \dots & \dots & 0_{3 \times 3} \\ 0_{3 \times 3} & [\mathbf{M}_2] & 0_{3 \times 3} & \dots & 0_{3 \times 3} \\ & & & \vdots & \\ 0_{3 \times 3} & 0_{3 \times 3} & \dots & 0_{3 \times 3} & [\mathbf{M}_N] \end{bmatrix}$ ,  $\mathbf{f}_E = [\mathbf{f}_{1,E}, \mathbf{f}_{2,E}, \dots, \mathbf{f}_{N,E}]^T$ , and  $\mathbf{f}_C = [\mathbf{f}_{1,C}, \mathbf{f}_{2,C}, \dots, \mathbf{f}_{N,C}]^T$ . By letting  $d_{ij,d}$  be the desired distance between  $m_i$  and  $m_j$ , and  $c_k$  be the  $k$ th constraint to be satisfied, the function of distance constraints can be expressed as follows from figure 3.1.

$$c_k = \|\mathbf{r}_i - \mathbf{r}_j\| - d_{ij,d} \quad (3.6)$$

where  $k$  is the number of constraint being considered, and  $k = 1, 2, \dots, k_f$ . Then, total number of constraints considered in a formation is collectively described as follows.

$$\mathbf{c} = \begin{bmatrix} c_1(\mathbf{r}) \\ c_2(\mathbf{r}) \\ \vdots \\ c_k(\mathbf{r}) \\ \vdots \\ c_{k_f}(\mathbf{r}) \end{bmatrix} \quad (3.7)$$

A VRB is established when all the constraint functions are satisfied for all time, i.e.  $c_k = 0 \forall k = 1, 2, \dots, k_f$ . To ensure this, the necessary conditions,  $\dot{\mathbf{c}} = 0$  and



$\ddot{\mathbf{c}} = 0$ , need to be satisfied as well. By taking derivatives of equation (3.7) with respect to time, following equations can be obtained.

$$\dot{\mathbf{c}} = \frac{d}{dt}\mathbf{c} = \frac{\partial \mathbf{c}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} = \mathbf{J}\dot{\mathbf{r}} = 0 \quad (3.8)$$

$$\ddot{\mathbf{c}} = \frac{d}{dt}\mathbf{J}\dot{\mathbf{r}} = \left\{ \frac{d}{dt}\mathbf{J} \right\} \dot{\mathbf{r}} + \mathbf{J} \left\{ \frac{d}{dt}\dot{\mathbf{r}} \right\} = \dot{\mathbf{J}}\dot{\mathbf{r}} + \mathbf{J}\ddot{\mathbf{r}} = 0 \quad (3.9)$$

where  $\mathbf{J} = \left[ \frac{\partial \mathbf{c}}{\partial \mathbf{r}} \right]$  is the constraint matrix [11], the rigidity matrix [21], or simply Jacobian matrix. Let us name equation (3.8) a constraint velocity, and equation (3.9) a constraint acceleration [11]. First, let us look at equation (3.9).

$\ddot{\mathbf{c}} = 0$  is satisfied when

$$\dot{\mathbf{J}}\dot{\mathbf{r}} = -\mathbf{J}\ddot{\mathbf{r}} \quad (3.10)$$

is true. Moreover, from equation (3.5),  $\ddot{\mathbf{r}}$  can be expressed as follows.

$$\ddot{\mathbf{r}} = \mathbf{M}^{-1}(\mathbf{f}_E + \mathbf{f}_C) \quad (3.11)$$

By substituting equation (3.11) into equation (3.10), we get

$$\dot{\mathbf{J}}\dot{\mathbf{r}} = -\mathbf{J}\mathbf{M}^{-1}(\mathbf{f}_E + \mathbf{f}_C) \quad (3.12)$$

By solving above equation for  $\mathbf{f}_C$ , constraint force is expressed as follows.

$$\mathbf{f}_C = [\mathbf{J}\mathbf{M}^{-1}]^{-1} \left[ -\dot{\mathbf{J}}\dot{\mathbf{r}} - \mathbf{J}\mathbf{M}^{-1}\mathbf{f}_E \right] \quad (3.13)$$

This expression of the constraint force equation (3.13) does not necessarily satisfy the constraint velocity, equation (3.8). To ensure equation (3.8) along with

equation (3.9), d'Alembert's principle of virtual work, which states work done to the system by the constraint force is zero [14], is applied.

$$\delta \mathbf{W}_C = \mathbf{f}_C \cdot \delta \mathbf{r} = 0 \quad (3.14)$$

To ensure this [11],

$$\mathbf{f}_C \cdot \delta \dot{\mathbf{r}} = 0 \quad (3.15)$$

Now, from equation (3.8), and equation (3.15), we get

$$\mathbf{J} \delta \dot{\mathbf{r}} = \mathbf{f}_C \cdot \delta \dot{\mathbf{r}} = 0 \quad (3.16)$$

From equation (3.16), the constraint force  $\mathbf{f}_C$  can be expressed as follows.

$$\mathbf{f}_C = \mathbf{J}^T \boldsymbol{\lambda} \quad (3.17)$$

where  $\boldsymbol{\lambda}$  is Lagrange multiplier. By substituting the constraint force, equation (3.13), which was derived from the constraint acceleration into above, we get the following equation.

$$\mathbf{J}^T \boldsymbol{\lambda} = [\mathbf{J} \mathbf{M}^{-1}]^{-1} [-\dot{\mathbf{J}} \dot{\mathbf{r}} - \mathbf{J} \mathbf{M}^{-1} \mathbf{f}_E] \quad (3.18)$$

By solving equation (3.18) for the Lagrange multiplier,

$$\boldsymbol{\lambda} = [\mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T]^{-1} [-\dot{\mathbf{J}} \dot{\mathbf{r}} - \mathbf{J} \mathbf{M}^{-1} \mathbf{f}_E] \quad (3.19)$$

By substituting equation (3.19) back into equation (3.17), the expression for constraint force that takes both constraint velocity and constraint acceleration into account is obtained as follows.

$$\mathbf{f}_C = \mathbf{J}^T [\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T]^{-1} [-\dot{\mathbf{J}}\dot{\mathbf{r}} - \mathbf{J}\mathbf{M}^{-1}\mathbf{f}_E] \quad (3.20)$$

### 3.2 Constraint Force Stabilization

As it appears in the constraint force expression, equation (3.20), stabilization is not explicitly embedded. Therefore if the initial conditions do not satisfy the constraint functions, the solution can diverge over time [8]; moreover, choosing initial conditions that satisfies the constraint functions becomes tedious and difficult as the number of agents increases [11]. Thus, utilization of feedback in the constraint functions must be considered. For the feedback control law to enforce these constraints, the Baumgarte stabilization technique [22], is employed. The Baumgarte stabilization technique is often used for numerical stabilization of multi-body constrained systems [8]. By implementing a PID-like control law derived from this Baumgarte stabilization technique, the constraint force is synthesized as follows. Similar work can be found in [7, 8, 12].

For holonomic constraint whose function is only in terms of position and time, the Baumgarte defines the system subjected to a constraint that is linear with respect to the acceleration as follows.

$$\ddot{\mathbf{c}} + 2\alpha\dot{\mathbf{c}} + \beta^2\mathbf{c} = 0 \quad (3.21)$$

where  $\alpha$  and  $\beta$  are constants to begin with and normally  $\alpha = \beta$ . From equation (3.9),  $\ddot{\mathbf{c}} = \dot{\mathbf{J}}\dot{\mathbf{r}} + \mathbf{J}\ddot{\mathbf{r}}$ , by substituting it into equation (3.21), we get

$$\dot{\mathbf{J}}\dot{\mathbf{r}} + \mathbf{J}\ddot{\mathbf{r}} + 2\alpha\dot{\mathbf{c}} + \beta^2\mathbf{c} = 0 \quad (3.22)$$

which can be modified to the following form.

$$\mathbf{J}\ddot{\mathbf{r}} = -\dot{\mathbf{J}}\dot{\mathbf{r}} - 2\alpha\dot{\mathbf{c}} - \beta^2\mathbf{c} \quad (3.23)$$

By substituting the expression of  $\ddot{\mathbf{r}}$  into equation (3.23), we get

$$\mathbf{JM}^{-1}(\mathbf{f}_E + \mathbf{f}_C) = -\dot{\mathbf{J}}\dot{\mathbf{r}} - 2\alpha\dot{\mathbf{c}} - \beta^2\mathbf{c} \quad (3.24)$$

Then, solving equation (3.24) for the constraint force, the following form of constraint force is obtained.

$$\mathbf{f}_C = [\mathbf{JM}^{-1}]^{-1} \left[ -\mathbf{JM}^{-1}\mathbf{f}_E - \dot{\mathbf{J}}\dot{\mathbf{r}} - 2\alpha\dot{\mathbf{c}} - \beta^2\mathbf{c} \right] \quad (3.25)$$

From here, d'Alembert's principle is utilized as it was done in the previous section. By following the same procedure, the following form for constraint force that includes the Baumgarte stabilization technique is obtained.

$$\mathbf{f}_C = [\mathbf{JM}^{-1}\mathbf{J}^T]^{-1} \left[ -\mathbf{JM}^{-1}\mathbf{f}_E - \dot{\mathbf{J}}\dot{\mathbf{r}} - 2\alpha\dot{\mathbf{c}} - \beta^2\mathbf{c} \right] \quad (3.26)$$

Now, by synthesizing an integral control, the constraint force with a PID-like control law derived from the Baumgarte stabilization technique is formed as follows.

$$\mathbf{f}_C = [\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T]^{-1} \left[ -\mathbf{J}\mathbf{M}^{-1}\mathbf{f}_E - \dot{\mathbf{J}}\dot{\mathbf{r}} - 2\alpha\dot{\mathbf{c}} - \beta^2\mathbf{c} - \gamma \int_0^t \mathbf{c} d\tau \right] \quad (3.27)$$

where  $\gamma$  is a gain to be tuned to an appropriate value. This modified constraint force is employed in all formations simulated in this paper.

## Chapter 4

### CONSTRAINT FORCE VALIDATION VIA SIMULATIONS

We have developed the equation for the constraint force as well as the 6-DOF governing equations of motion of a multiagent system. In this section, we will first try to validate the establishment of VRB formation by using the constraint force we developed in the previous section.

By implementing equation (3.3) along with equation (3.27) in Simulink, the following simulink model was developed.

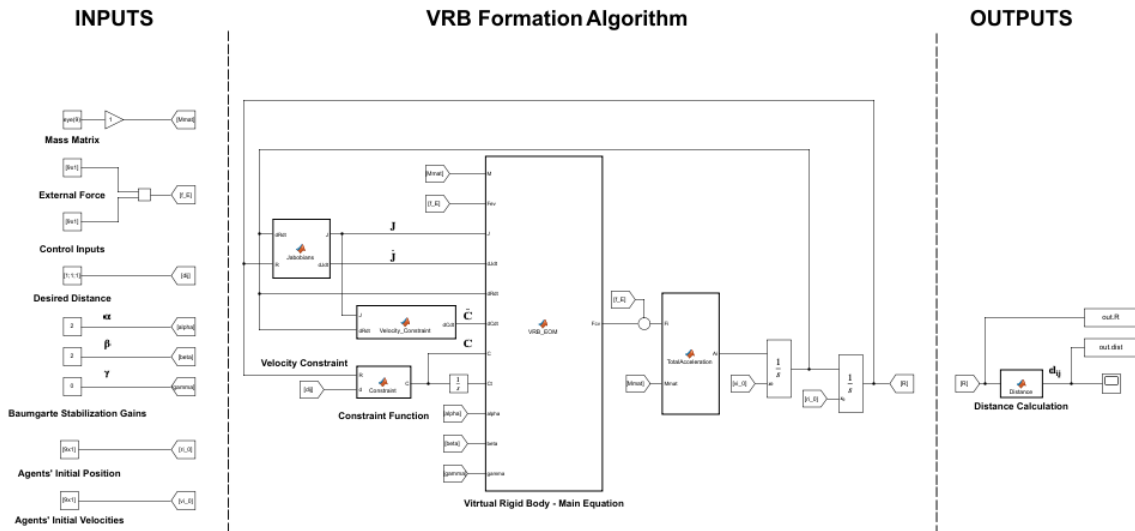


Figure 4.1. Simulink model for formation establishment.

The architecture of the Simulink model is illustrated below.

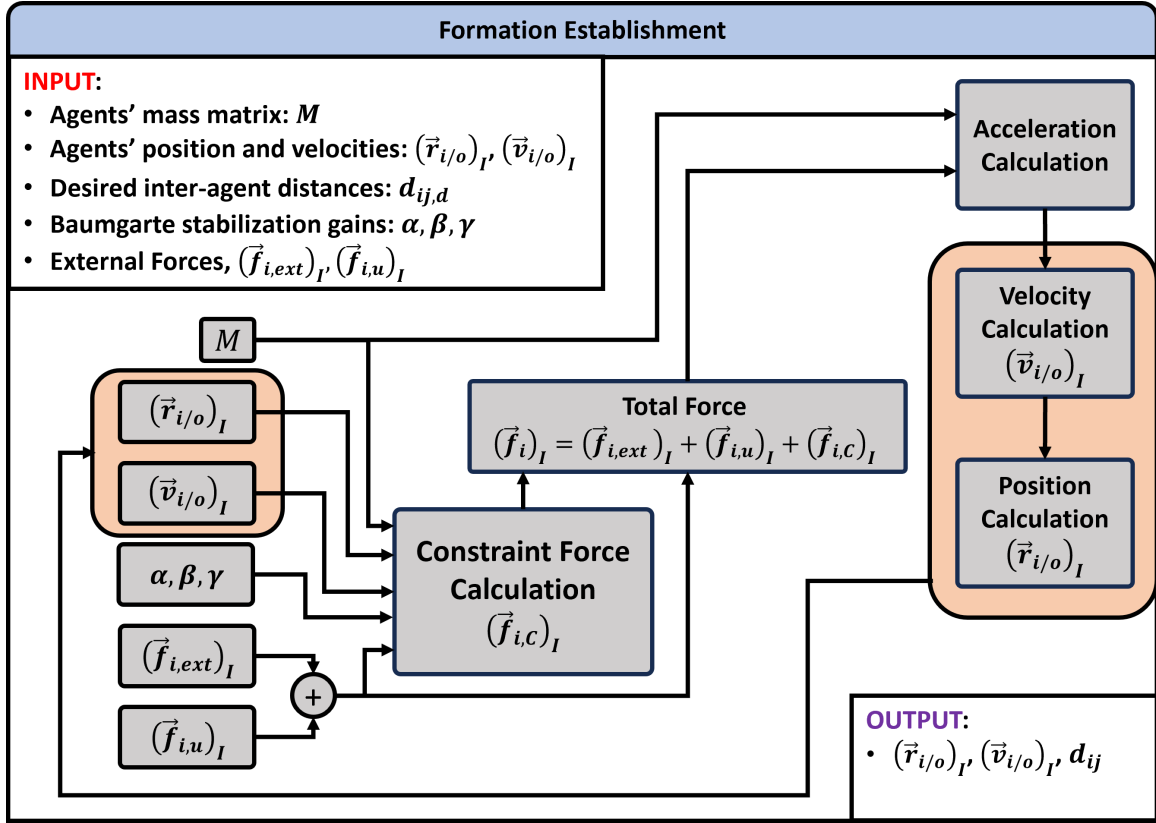


Figure 4.2. Simulink model architecture for formation establishment.

Following common conditions are applied for the simulations in this section.

Input specifications	
Input Description	Values
Number of Agents	$N = 3$
Agent's mass	$m_i = 1kg \forall i = 1, 2, 3$
Initial position	$\mathbf{r}_{1,0} = [1, 6, 3]^T m, \mathbf{r}_{2,0} = [8, 3, 3]^T m, \mathbf{r}_{3,0} = [7, 6, 3]^T m$
Initial velocity	$\mathbf{v}_{i,0} = [0, 0, 0]^T m/s \forall i = 1, 2, 3$
External force	$\mathbf{f}_{i,ext} = [0, 0, -m_i g]^T N$ (gravitational force)
Input force	$\mathbf{f}_{i,u} = [0, 0, m_i g]^T N$ (hovering force)

Table 4.1. Formation establishment input specifications

While there are other source of external forces from the surrounding environment, let us consider only the gravitational force to keep the generality of this research. For the control inputs applied to each agent, let us first apply the force required to hover for a VTOL capable vehicle. If the algorithm works perfectly, it is expected that the position of the center of mass stays still while the each point mass moves to establish a formation specific to the desired distances between agents.

From this section onward, let us consider a three-agent case to simplify the visualization. In this section, we will try to establish a triangle formation, a linear formation, and the reconfiguration of the formations between them. To develop these formations, we will consider the following constraint relationship between agents.

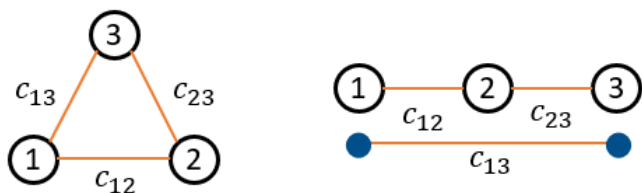


Figure 4.3. Three-agent cases constraint graphs. (Left): Equilateral triangle formation, (Right): Linear formation. Orange lines - applied constraints.

For equilateral triangle formation, following distances between agents are considered.

$$\begin{bmatrix} d_{12,d} & d_{13,d} & d_{23,d} \end{bmatrix} = \begin{bmatrix} 4 & 4 & 4 \end{bmatrix} m$$

As a result of simulation, following outcomes were obtained.



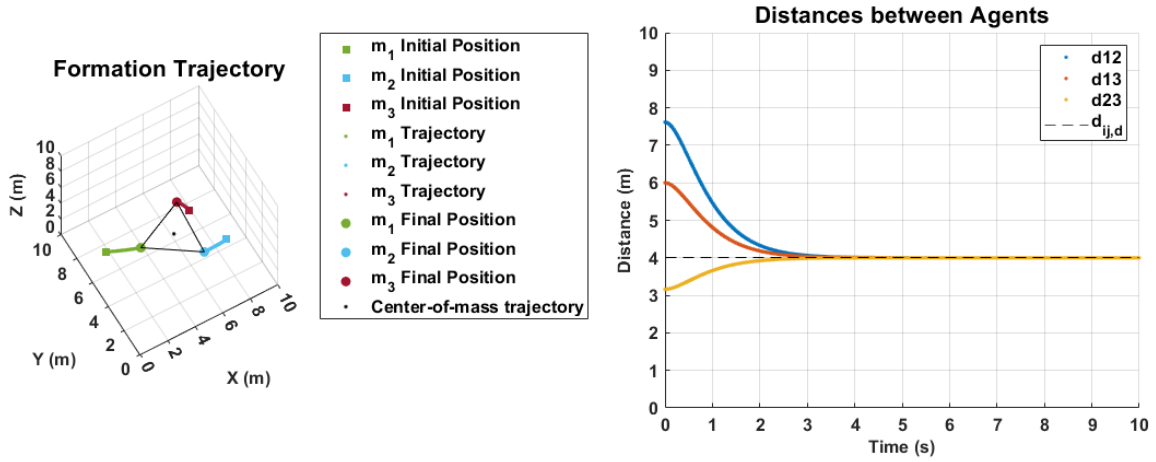


Figure 4.4. Equilateral triangle formation establishment.

As figure 4.4 shows, the desired distances between agents are achieved and stabilized; thus, the expected formation is established and maintained for the simulation period. Moreover, the formation center of mass was held stationary throughout the simulation as expected. Now, let us consider the linear formation. For the linear formation, we consider the following distances between agents.

$$\begin{bmatrix} d_{12,d} & d_{13,d} & d_{23,d} \end{bmatrix} = \begin{bmatrix} 4 & 8 & 4 \end{bmatrix} m$$

From the desired distances, the final formation is expected to be a linear formation. As a result of simulation, the following outcomes were obtained.

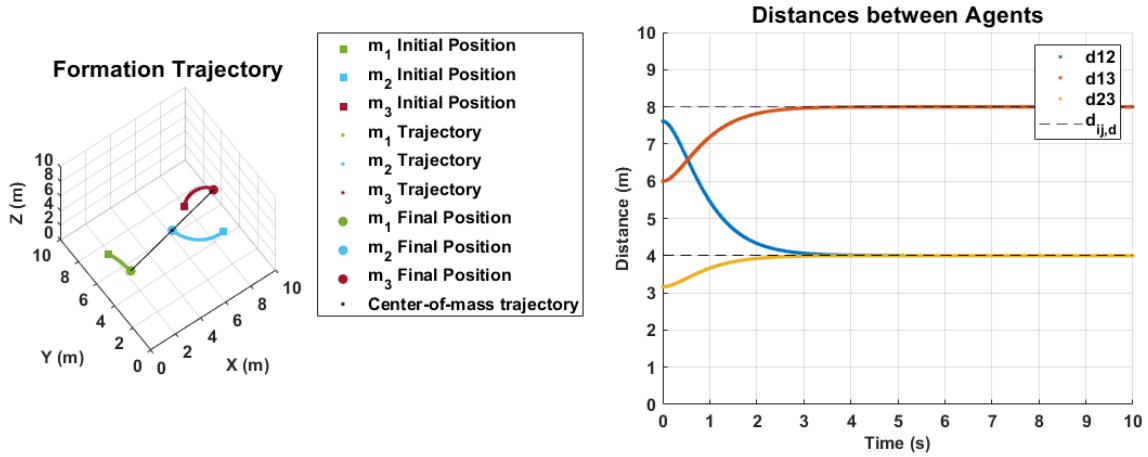


Figure 4.5. Three-agent linear formation.

As figure 4.5 shows, the linear formation derived from the desired distances is established and maintained while the position of the formation center of mass is held constant. Now, let us consider another situation where the formation switches its configuration from equilateral triangle to linear and vice-versa.

Following inputs are considered.

$$t = [0 : 10]; \begin{bmatrix} d_{12,d} & d_{13,d} & d_{23,d} \end{bmatrix} = \begin{bmatrix} 4 & 4 & 4 \end{bmatrix} m \text{ (Triangular Formation)}$$

$$t = [10 : 20]; \begin{bmatrix} d_{12,d} & d_{13,d} & d_{23,d} \end{bmatrix} = \begin{bmatrix} 4 & 8 & 4 \end{bmatrix} m \text{ (Linear Formation)}$$

$$t = [20 : 30]; \begin{bmatrix} d_{12,d} & d_{13,d} & d_{23,d} \end{bmatrix} = \begin{bmatrix} 4 & 4 & 4 \end{bmatrix} m \text{ (Triangle Formation)}$$

From the inputs, it is expected that the formation reconfigurations between triangle formation and linear formation occurs at 10 seconds period. As a result of simulation, following results were obtained.

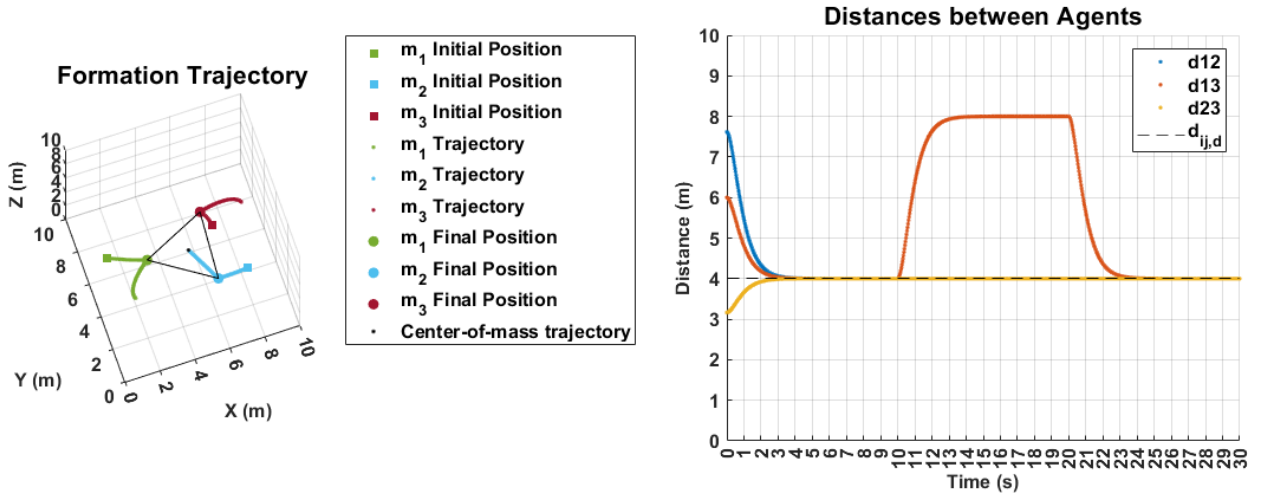


Figure 4.6. Three-agent formation reconfiguration.

As figure 4.6 shows, after the step input for the linear formation is applied, the formation reconfiguration from triangular to linear started, and the formation is, again, reconfigured from linear to triangular after  $t = 20$  seconds. For each stage, desired formations were established and maintained. Now, in addition to the input for hovering, let us apply another input that translates the formation. The following input is applied.

$$\mathbf{f}_{i,u} = \begin{bmatrix} 0.02 & 0.02 & mg & 0.02 & 0.02 & mg & 0.02 & 0.02 & mg \end{bmatrix}^T N \quad (4.1)$$

This input is applied while the triangle-to-linear formation reconfiguration is underway. As a result of a simulation, the following results were obtained.

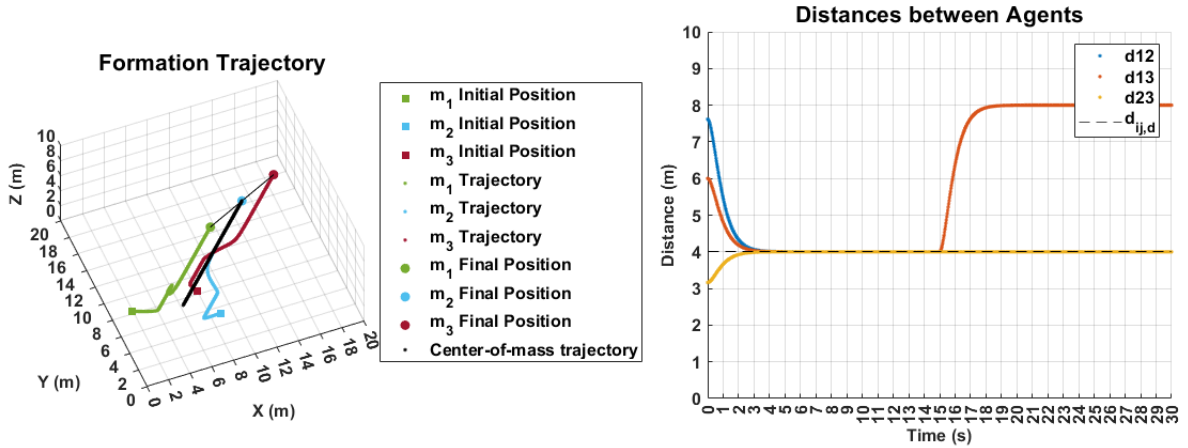


Figure 4.7. Three-agent triangle-to-linear formation under translational motion.

As figure 4.7 shows, translation of the formation is achieved as the formation shape is reconfigured from triangular formation to linear formation. For both formations, desired formations derived from the distance constraints are established and maintained. From these simulation results, it can be concluded that this VRB formation framework is viable for multiple formation situations including formation reconfigurations. Formation establishments with different numbers of agents are also simulated and shown in Appendix A.

## Chapter 5

### FORMATION BODY FRAME AND INPUT DECOUPLING

As it is shown in the development of the equations of motion, the system has a body frame attached to itself. Therefore, we wish to come up with a method to attach a body frame to a VRB formation. Attachment of the body frame associated with the formation is performed by utilizing the geometric features of the formation. Moreover, inputs need to be distributed to each agent precisely so that the overall motion of the VRB due to the motion of each agent represents the desired behavior. Therefore, the input decoupling method is also discussed in this section.

#### 5.1 Body Frame Attachment

The body frame of the VRB formation can be attached with the following steps.

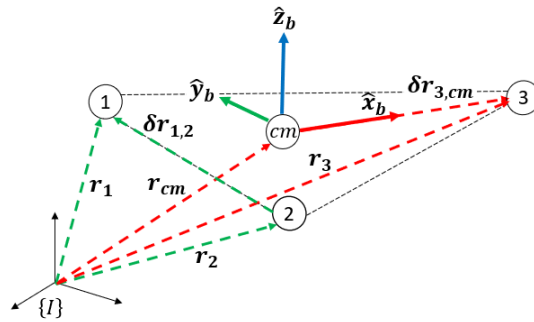


Figure 5.1. Formation with body frame.

Once a formation is established, relative position vectors that are parallel to the desired body frame can be found in the inertial frame. For example,  $\hat{x}_B$  axis can be found from  $\delta\mathbf{r}_{3,cm}$ , and  $\hat{y}_B$  axis can be found from a vector perpendicular to  $\delta\mathbf{r}_{3,cm}$ , such as  $\delta\mathbf{r}_{1,2}$ . By performing a cross product of these two vectors, the last axis can be determined.

$$\begin{aligned}
(\delta\mathbf{r}_{1,2})_I &= (\mathbf{r}_1)_I - (\mathbf{r}_2)_I \\
(\delta\mathbf{r}_{3,cm})_I &= (\mathbf{r}_3)_I - (\mathbf{r}_{cm})_I \\
(\mathbf{r}_{z,cm})_I &= (\delta\mathbf{r}_{3,cm})_I \times (\delta\mathbf{r}_{1,2})_I
\end{aligned} \tag{5.1}$$

By obtaining unit vectors for them, the body frame axes can be initialized.

$$\begin{aligned}
\hat{\mathbf{x}}_B &= \frac{\delta\mathbf{r}_{3,cm}}{\|\delta\mathbf{r}_{3,cm}\|} \\
\hat{\mathbf{y}}_B &= \frac{\delta\mathbf{r}_{2,1}}{\|\delta\mathbf{r}_{2,1}\|} \\
\hat{\mathbf{z}}_B &= \frac{\mathbf{r}_{z,cm}}{\|\mathbf{r}_{z,cm}\|}
\end{aligned} \tag{5.2}$$

After obtaining the body axes in the inertial frame, the transformation matrix from inertial to body frame can be constructed. By using an inertial-to-body transformation matrix from a 3-2-1 sequence, we can get the initial orientation of the formation based on the body frame attached to it.

$$\begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \\ \hat{\mathbf{z}} \end{bmatrix}_B = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_I \tag{5.3}$$

Then,

$$\begin{aligned}
\phi_0 &= \tan^{-1} \left( \frac{C_{23}}{C_{33}} \right) \\
\theta_0 &= -\sin^{-1} (C_{13}) \\
\psi_0 &= \tan^{-1} \left( \frac{C_{12}}{C_{11}} \right)
\end{aligned} \tag{5.4}$$

By using the initial Euler angles obtained from equation (5.4), Quaternions can be initialized as follows.

$$\begin{bmatrix} k_{0,0} \\ k_{1,0} \\ k_{2,0} \\ k_{3,0} \end{bmatrix} = \begin{bmatrix} \cos(\frac{\phi_0}{2}) \cos(\frac{\theta_0}{2}) \cos(\frac{\psi_0}{2}) + \sin(\frac{\phi_0}{2}) \sin(\frac{\theta_0}{2}) \sin(\frac{\psi_0}{2}) \\ \sin(\frac{\phi_0}{2}) \cos(\frac{\theta_0}{2}) \cos(\frac{\psi_0}{2}) - \cos(\frac{\phi_0}{2}) \sin(\frac{\theta_0}{2}) \sin(\frac{\psi_0}{2}) \\ \cos(\frac{\phi_0}{2}) \sin(\frac{\theta_0}{2}) \cos(\frac{\psi_0}{2}) + \sin(\frac{\phi_0}{2}) \cos(\frac{\theta_0}{2}) \sin(\frac{\psi_0}{2}) \\ \cos(\frac{\phi_0}{2}) \cos(\frac{\theta_0}{2}) \sin(\frac{\psi_0}{2}) - \sin(\frac{\phi_0}{2}) \sin(\frac{\theta_0}{2}) \cos(\frac{\psi_0}{2}) \end{bmatrix} \tag{5.5}$$

While some of these processes are only applicable for specific formation shape, the body frame of a formation with more or less agents or a different shape can be generated in a similar manner.

## 5.2 Input Decoupling

After the input is calculated with methods one wishes to use, the control input needs to be distributed precisely and properly to each agent so that the overall motion of VRB due to the motion of each agent represents the desired behavior. From force and torque input, the following calculation can be performed.

$$\begin{aligned}
(\mathbf{f}_{cm})_B &= \sum_{i=1}^N (\mathbf{f}_i)_B \\
(\boldsymbol{\tau}_{cm})_B &= \sum_{i=1}^N (\boldsymbol{\tau}_{cm/i})_B = \sum_{i=1}^N [(\delta \mathbf{r}_{i/cm})_B]^\times (\mathbf{f}_i)_B
\end{aligned} \tag{5.6}$$

Then, equation (5.6) can be expressed, in a matrix form, as follows.

$$\begin{bmatrix} I_{3 \times 3} & \dots & \dots & \dots & I_{3 \times 3} \\ [(\delta \mathbf{r}_{1,cm})_B]^\times & \mathbf{0}_{3 \times 3} & \dots & \dots & \mathbf{0}_{3 \times 3} \\ 0_{3 \times 3} & [(\delta \mathbf{r}_{2,cm})_B]^\times & \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} \\ \vdots & & \vdots & & \vdots \\ 0_{3 \times 3} & \dots & \dots & \mathbf{0}_{3 \times 3} & [(\delta \mathbf{r}_{N,cm})_B]^\times \end{bmatrix} \begin{bmatrix} (\mathbf{f}_1)_B \\ (\mathbf{f}_2)_B \\ \vdots \\ \vdots \\ (\mathbf{f}_N)_B \end{bmatrix} = \begin{bmatrix} (\mathbf{f}_{cm})_B \\ (\boldsymbol{\tau}_{cm,1})_B \\ (\boldsymbol{\tau}_{cm,2})_B \\ \vdots \\ (\boldsymbol{\tau}_{cm,N})_B \end{bmatrix} \quad (5.7)$$

By abbreviating the matrix in equation (5.7) as  $\mathbf{H}$ , the control input on each agent can be calculated as follows.

$$\begin{bmatrix} (\vec{\mathbf{f}}_1)_B \\ (\vec{\mathbf{f}}_2)_B \\ \vdots \\ \vdots \\ (\vec{\mathbf{f}}_N)_B \end{bmatrix} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \begin{bmatrix} (\vec{\mathbf{f}}_{cm})_B \\ (\vec{\boldsymbol{\tau}}_{cm,1})_B \\ (\vec{\boldsymbol{\tau}}_{cm,2})_B \\ \vdots \\ (\vec{\boldsymbol{\tau}}_{cm,N})_B \end{bmatrix} \quad (5.8)$$

$$\text{where } \mathbf{H} = \begin{bmatrix} I_{3 \times 3} & \dots & \dots & \dots & I_{3 \times 3} \\ [(\delta \mathbf{r}_{1,cm})_B]^\times & \mathbf{0}_{3 \times 3} & \dots & \dots & \mathbf{0}_{3 \times 3} \\ 0_{3 \times 3} & [(\delta \mathbf{r}_{2,cm})_B]^\times & \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} \\ \vdots & & \vdots & & \vdots \\ 0_{3 \times 3} & \dots & \dots & \mathbf{0}_{3 \times 3} & [(\delta \mathbf{r}_{N,cm})_B]^\times \end{bmatrix}$$



## Chapter 6

### SYSTEM SYNTHESIS AND SIMULATION

In this section, all the components developed in the previous sections will be combined into one framework and simulated under applications of various inputs to check the feasibility of the synthesized system.

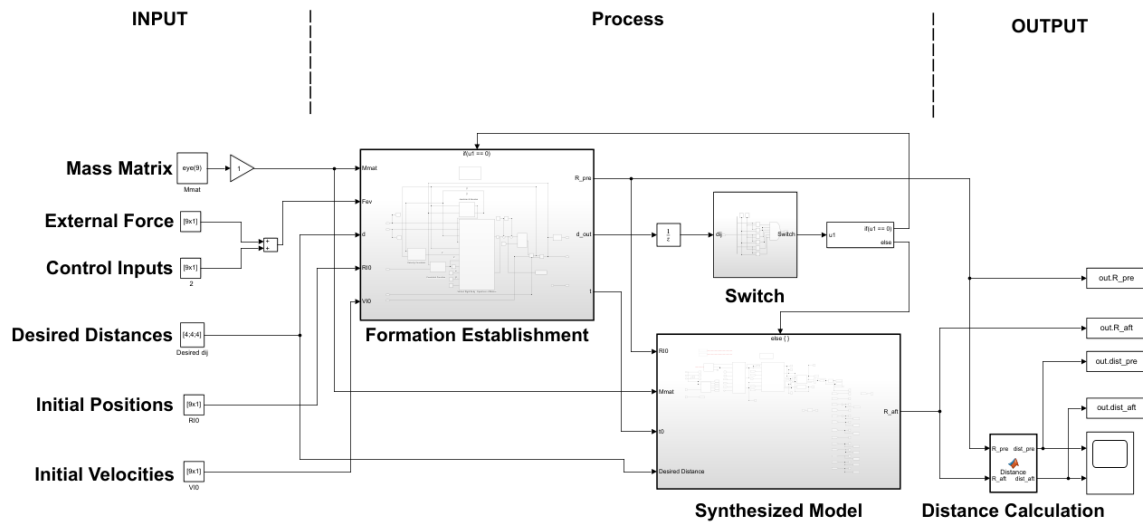


Figure 6.1. Synthesized Simulink model.

The architecture of the simulink model is explained below.

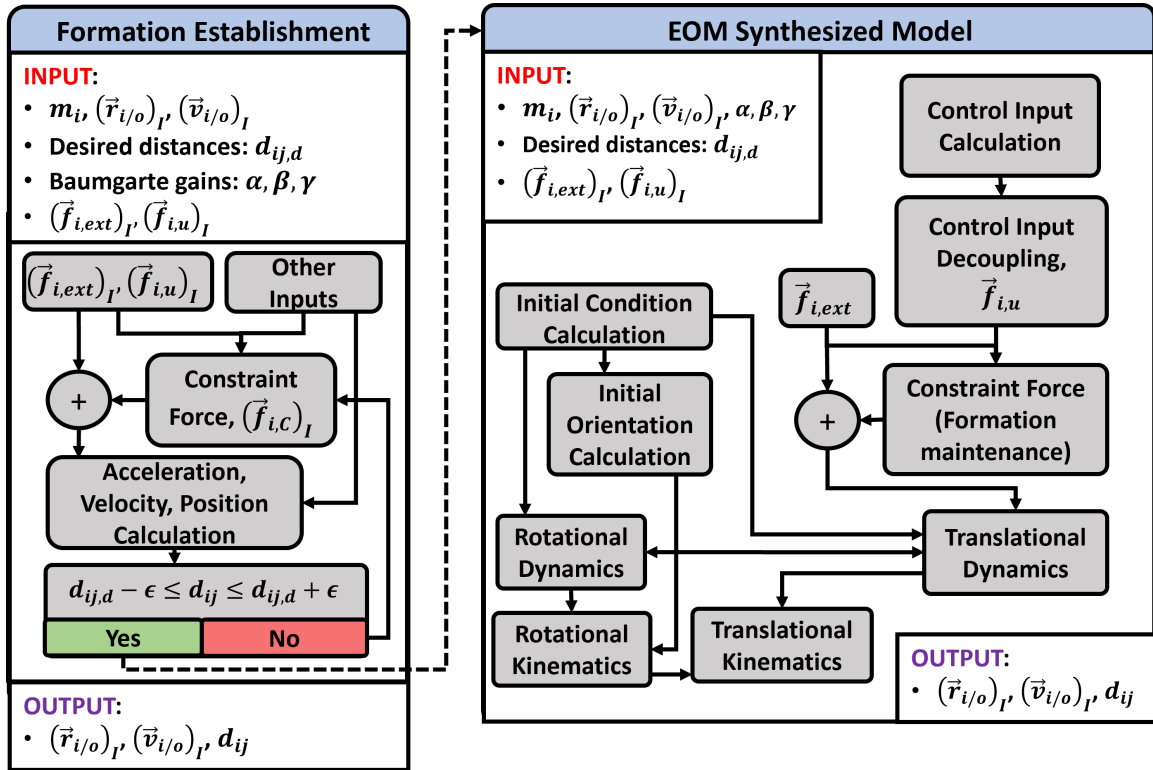


Figure 6.2. Synthesized Simulink model architecture.

## 6.1 Straight Line Trajectory

We will check the synthesized system with a linear motion in this section. The following values were applied as inputs.

Input specifications	
Input Description	Values
Number of Agents	$N = 3$
Agent's mass	$m_i = 1kg \forall i = 1, 2, 3$
Desired distances	$[d_{12,d}, d_{13,d}, d_{23,d}] = [4, 4, 4]m$
Initial position	$\mathbf{r}_{1,0} = [1, 6, 3]^T m, \mathbf{r}_{2,0} = [8, 3, 3]^T m, \mathbf{r}_{3,0} = [7, 6, 3]^T m$
Initial velocity	$\mathbf{v}_{i,0} = [0, 0, 0]^T m/s \forall i = 1, 2, 3$
External force	$\mathbf{f}_{cm,ext} = [0, 0, -m_{cm}g]^T N$ (gravitational force)
Input force	$\mathbf{f}_{cm,u} = [\mathbf{T}]_{B,0}^T [0.2, 0, 0]^T + [0, 0, m_{cm}g]^T (N)$
Input torque	$\boldsymbol{\tau}_{cm} = [0, 0, 0]^T N \cdot m$
Stabilization gains	$\alpha = 2, \beta = 2, \gamma = 0$

Table 6.1. Formation linear motion input specifications

From the inputs shown above, it is expected that the formation translates in the  $\hat{x}_{B,0}$  direction without changing its altitude. Moreover, since no torque is applied to the formation, angular rates about all body axes are expected to be zero, which should result in constant Euler angles.

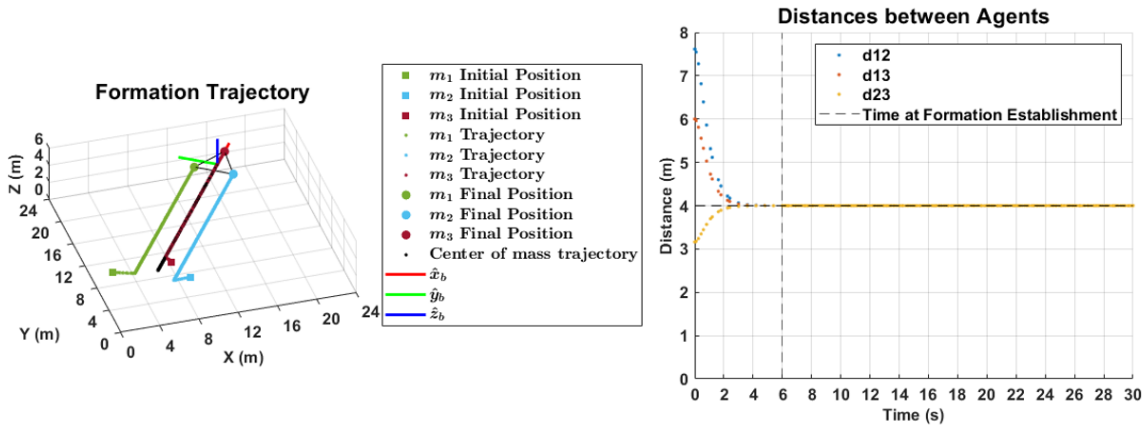


Figure 6.3. Formation under the constant input - Linear motion.

As figure 6.3 shows, the formation is translated in its  $\hat{x}_{B,0}$  direction as the formation is established. Moreover, as it was expected, angular rates of the formation stayed at zero, and thus the Euler angles of the formation were held constant. Now, let us look at the behaviors of the formation under other inputs, including torque.

## 6.2 Rolling Maneuver

In this section, rolling maneuver is simulated. The following torque input is applied while other inputs are the same as the previous case.

Input specifications	
Input Description	Values
Input torque	$\boldsymbol{\tau}_{cm} = [0.5, 0, 0]^T N \cdot m$
Stabilization gains	$\alpha = 2, \beta = 2, \gamma = 0$ (Formation Establishment)
Stabilization gains	$\alpha = 20, \beta = 20, \gamma = 40$ (Formation Maintenance)

Table 6.2. Formation roll motion input specifications

The gains for formation maintenance were appropriately chosen since steady-state error in the formation is observed as rotational motion is applied to the formation. As a result of simulation, the following outcomes were obtained.

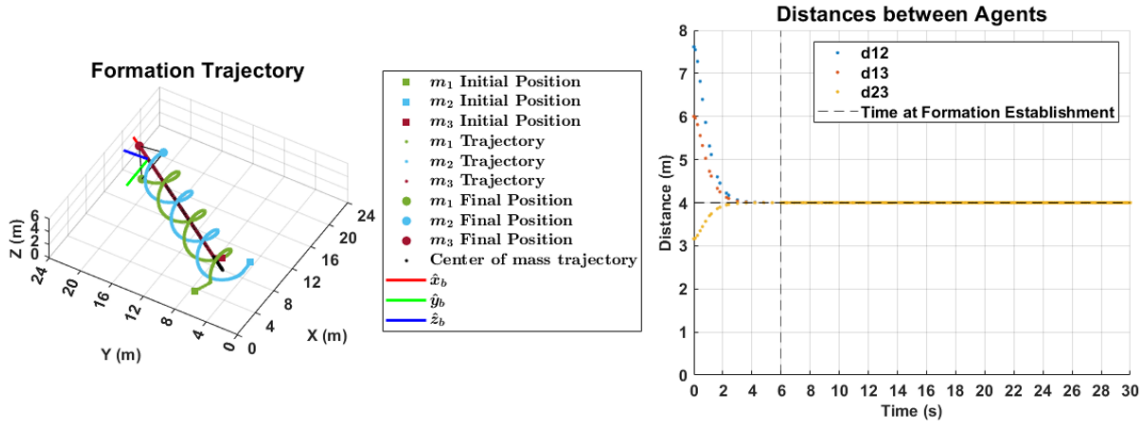


Figure 6.4. Formation under the constant input - Rolling motion.

As you can see from figure 6.4, the rolling maneuver was initiated as the triangle formation was established. During the rolling maneuver, formation was kept intact for the whole simulation period.

### 6.3 Pitching Maneuver

In this section, a pitching maneuver is simulated. The following inputs are applied.

Input specifications	
Input Description	Values
Input torque	$\tau_{cm} = [0, 0.5, 0]^T N \cdot m$
Stabilization gains	$\alpha = 2, \beta = 2, \gamma = 0$ (Formation Establishment)
Stabilization gains	$\alpha = 20, \beta = 20, \gamma = 40$ (Formation Maintenance)

Table 6.3. Formation pitch motion input specifications

As a result, following simulation results were obtained.

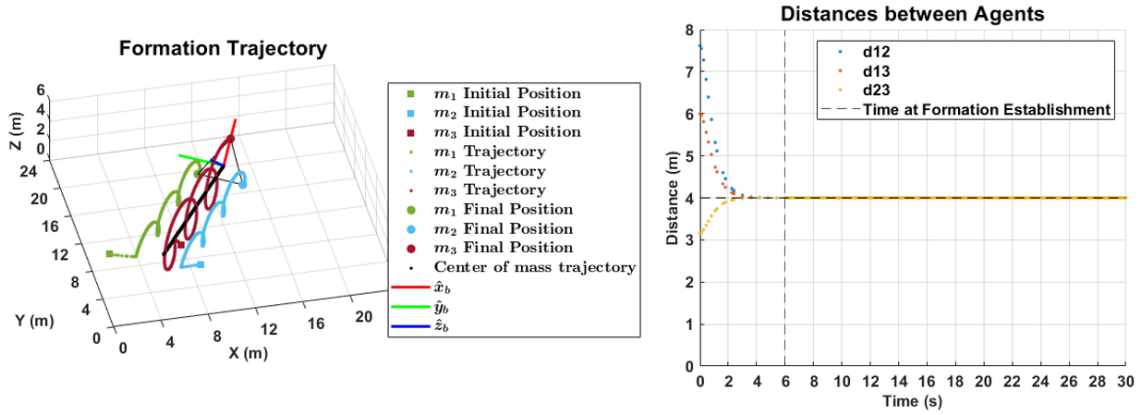


Figure 6.5. Formation under the constant input - Pitching motion.

As figure 6.5 shows, the pitching motion was initiated as the triangle formation was established. While the formation is in the pitch motion, the formation center of mass moved linearly, and the VRB formation was maintained.

#### 6.4 Yawing Maneuver

In this section, a yawing maneuver is simulated. The following inputs are applied.

Input specifications	
Input Description	Values
Input torque	$\tau_{cm} = [0, 0, 0.5]^T N \cdot m$
Stabilization gains	$\alpha = 2, \beta = 2, \gamma = 0$ (Formation Establishment)
Stabilization gains	$\alpha = 20, \beta = 20, \gamma = 40$ (Formation Maintenance)

Table 6.4. Formation yaw motion input specifications

As a result, following simulation results were obtained.

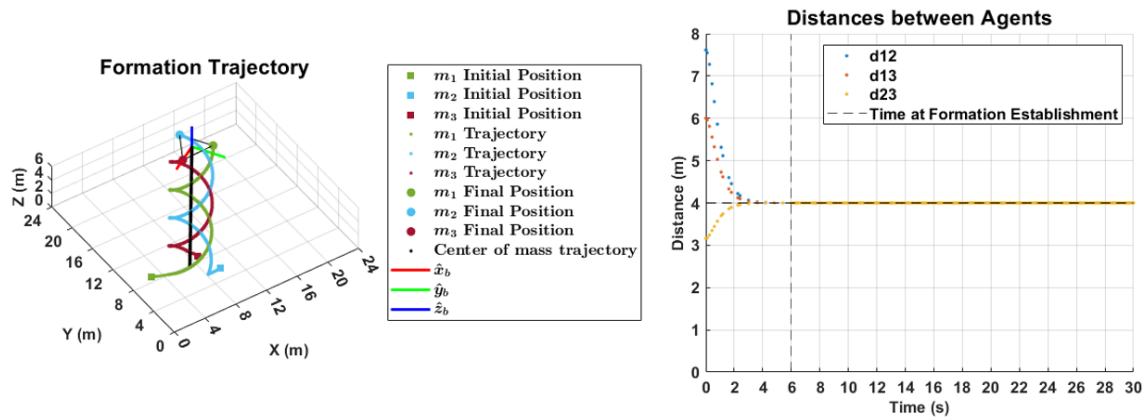


Figure 6.6. Formation under the constant input - Yawing motion.

As it can be seen from figure 6.6, the yawing motion was initiated as the triangle formation was established. Similar to the rolling and pitching simulations, the formation center of mass was translated linearly while the formation was maintained.

From the simulations performed in this section, it was shown that the system is viable under the application of multiple inputs and is capable of generating expected behaviors.

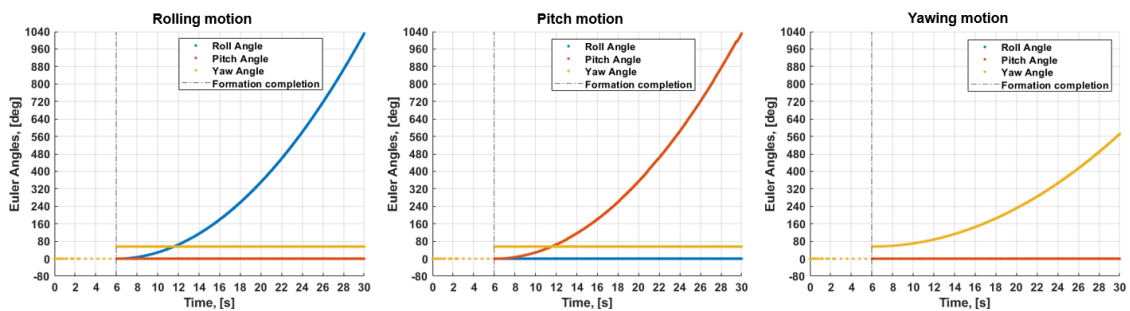


Figure 6.7. Results of Euler angles.

Moreover, as figure 6.7 shows, due to the synthesis of quaternions in the framework, singularity due from the Euler angles were avoided, and all-attitude capability was achieved. Therefore, we can conclude that the framework of the formation establishment, reconfiguration, orientation, and maintenance has been established. Now, let us see if the formation can achieve the desired behaviors under the inputs calculated from multiple guidance laws such as waypoint tracking and trajectory tracking.



## Chapter 7

### APPLICATION EXAMPLES

Motivated by the results of the previous section, we will attempt the integration of the multiple control input calculation methods, such as linear quadratic control method, to the framework and simulate it with in an environment including multiple key occasions. The environment considered in this section is shown below.

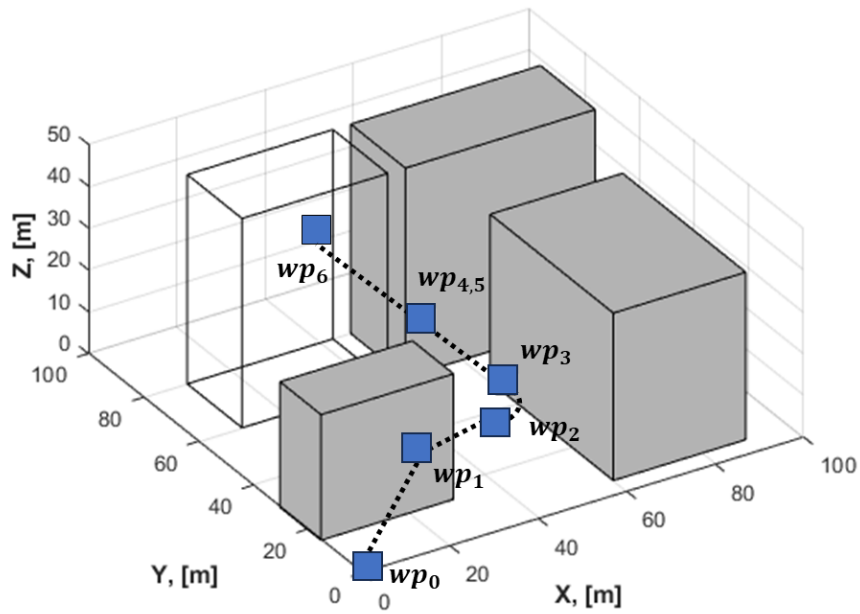


Figure 7.1. Environment setup.

Before considering the whole environment involving multiple waypoint, we will try to see if the framework is capable of performing single-waypoint flight.

## 7.1 Single Waypoint Simulations

In this section, let us consider a single-waypoint flight that requires control of the formation reorientation. Following simulation inputs and the way point conditions are considered.

Simulation Input and Waypoint Specifications	
Simulation Input	Values
Number of Agents	$N = 3$
Agent's mass	$m_i = 1.023kg \forall i = 1, 2, 3$
Desired distances	$d_{ij} = 4m \forall ij = 12, 13, 23$
Initial position	$\mathbf{r}_{1,0} = [1, 6, 3]^T m, \mathbf{r}_{2,0} = [8, 3, 3]^T m, \mathbf{r}_{3,0} = [7, 6, 3]^T m$
Initial velocity	$\mathbf{v}_{i,0} = [0, 0, 0]^T m/s \forall i = 1, 2, 3$
External force	$\mathbf{f}_{i,ext} = [0, 0, -m_i g]^T N$ (gravitational force)
Waypoint	Values
C.M. position	$\mathbf{r}_{cm} = [15, 15, 15]^T m$
C.M. velocity	$\mathbf{v}_{cm} = [0, 0, 0]^T m/s$
VRB orientation	$\boldsymbol{\sigma} = [0, 0, -90]^{oT}$
VRB angular rates	$\boldsymbol{\omega}_b = [0, 0, 0]^{oT}/s$

Table 7.1. Simulation specifications for single waypoint case

As candidate methods, closed-loop optimal control with Linear Quadratic Regulator (LQR), Local Voting Protocol, and trajectory tracking are considered.

### 7.1.1 Linear Quadratic Regulator

In this section, closed-loop control with minimum state and input energy is considered by utilizing a Hamiltonian approach. Detailed derivations of the method are well explained in [23]. For our formation framework, the dynamics of the VRB are described as follows. Please note that the size of vector is  $13 \times 1$  since the Quaternion is employed.

$$\left\{ \begin{array}{l}
\dot{x}_{cm} = u_{cm} \\
\dot{y}_{cm} = v_{cm} \\
\dot{z}_{cm} = w_{cm} \\
\dot{u}_{cm} = f_x/m_{cm} \\
\dot{v}_{cm} = f_y/m_{cm} \\
\dot{w}_{cm} = f_z/m_{cm} \\
\dot{\boldsymbol{\omega}}_B = [\mathbf{I}_{cm}]_B^{-1} \left\{ -\boldsymbol{\omega}_B \times [\mathbf{I}_{cm}]_B \boldsymbol{\omega}_B + (\boldsymbol{\tau}_{cm})_B \right\} \\
\dot{k}_0 = \frac{1}{2}(-pk_1 - qk_2 - rk_3) \\
\dot{k}_1 = \frac{1}{2}(pk_0 + rk_2 - qk_3) \\
\dot{k}_2 = \frac{1}{2}(pk_0 - rk_1 + pk_3) \\
\dot{k}_3 = \frac{1}{2}(rk_0 + qk_1 - pk_2)
\end{array} \right. \quad (7.1)$$

By linearizing the equations in equation (7.1), the linear state-space equations are easily obtained and can be expressed as below.

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}U \quad (7.2)$$

Now, the closed-loop optimal control inputs are calculated by utilizing the following components in the table below.

<b>Closed-Loop Optimal Control Components</b>	
<b>Components</b>	<b>Equations</b>
System Model	$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}$
Performance Index (min. state and input energy)	$J = \frac{1}{2}\mathbf{X}(T)\mathbf{S}(T)\mathbf{X}(T) + \frac{1}{2}\int_0^T (\mathbf{X}^T\mathbf{Q}\mathbf{X} + \mathbf{U}^T\mathbf{R}\mathbf{U})dt$
Ricatti Equation	$-\dot{\mathbf{S}} = \mathbf{A}^T\mathbf{S} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} + \mathbf{Q}$
Control Gain	$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}$

Table 7.2. Linear Quadratic Regulator Components

After implementing LQR into the formation framework within Simulink, the following simulation result was obtained.

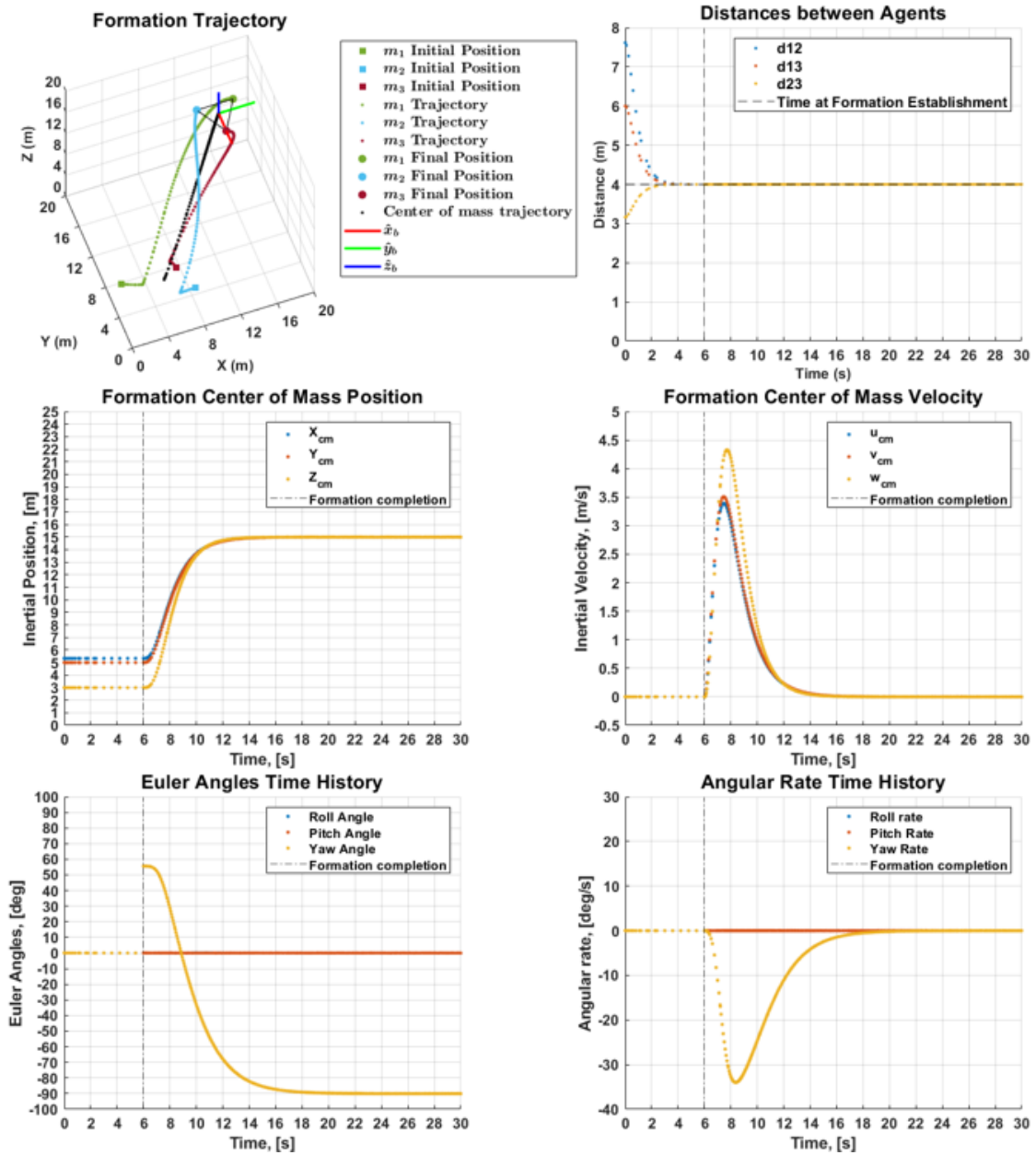


Figure 7.2. Single waypoint simulation results via LQR.

As figure 7.2 shows, as the triangle formation is established, the trajectory command is executed, and the formation center of mass as well as the attitude of the formation reached their desired values and stabilized. Thus, formation establishment,

formation orientation, and station-keeping are successful using the current framework and LQR. Now, as a supplemental outcome, let us check the control input applied to each agent.

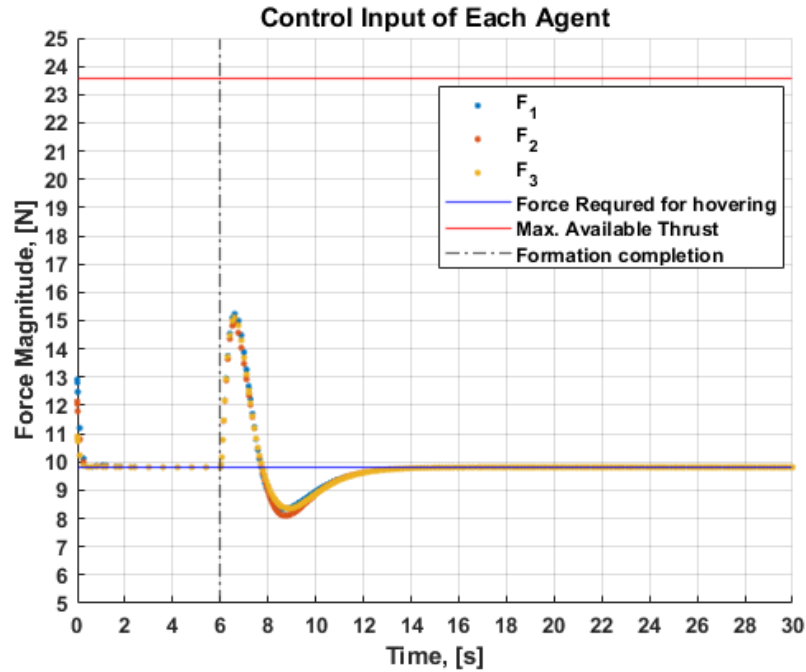


Figure 7.3. Single way point simulation results via LQR: Control Inputs.

According to [24], the ASL (Aerospace System Laboratory) quadcopter model approximately has maximum available thrust of 5.9 N for each rotor including thrust required for hovering. Therefore, the total available thrust for the quadcopter is approximately 23.6 N. As figure 7.3 shows, the control input applied to each agent is all well below the maximum available thrust. Since our agents are currently point masses, thrust required for each rotor is unknown. Therefore, when implementation of this framework to the physical system is considered, further investigation needs to be carried out.

### 7.1.2 Local Voting Protocol

Local voting protocol seeks to make the differences between one's state and those of its neighbors equal to zero [23]. This concept is often utilized to achieve the collective motion of a large group that tries to satisfy fundamental rules for copoperative flight, such as collision avoidance, velocity matching, and flock centering [23]. In this section, we will try to use this concept of local voting protocol to achieve waypoint conditions. Derivations of the equations utilized in this section are well explained in or easily derived from [23].

<b>Local Voting Protocol</b>	
<b>Consensus</b>	<b>Equations</b>
Position & Velocity	$(\mathbf{f}_{cm})_I = \mathbf{M}\dot{\mathbf{v}}_{cm} = \mathbf{M}\left\{a(\mathbf{r}_{wp} - \mathbf{r}_{cm}) + b(\mathbf{v}_{wp} - \mathbf{v}_{cm})\right\}$
Orientation & Angular Rate	$(\boldsymbol{\tau}_{cm})_B = c(\boldsymbol{\sigma}_{wp} - \boldsymbol{\sigma}_{cm}) + d\left\{(\boldsymbol{\omega}_{wp})_B - (\boldsymbol{\omega}_{cm})_B\right\}$

Table 7.3. Local Voting Protocol Components

The downside of this method is that this method is incompatible with Quaternions due to unequal size of the elements since Quaternions have four elements, i.e.  $\boldsymbol{\tau}_{cm}, \boldsymbol{\omega} \in \mathbb{R}^{3 \times 1}$  while  $\mathbf{k} \in \mathbb{R}^{4 \times 1}$ . After implementing the equation in Table 7.3 as control inputs with the desired way point conditions, the following simulation results were obtained.

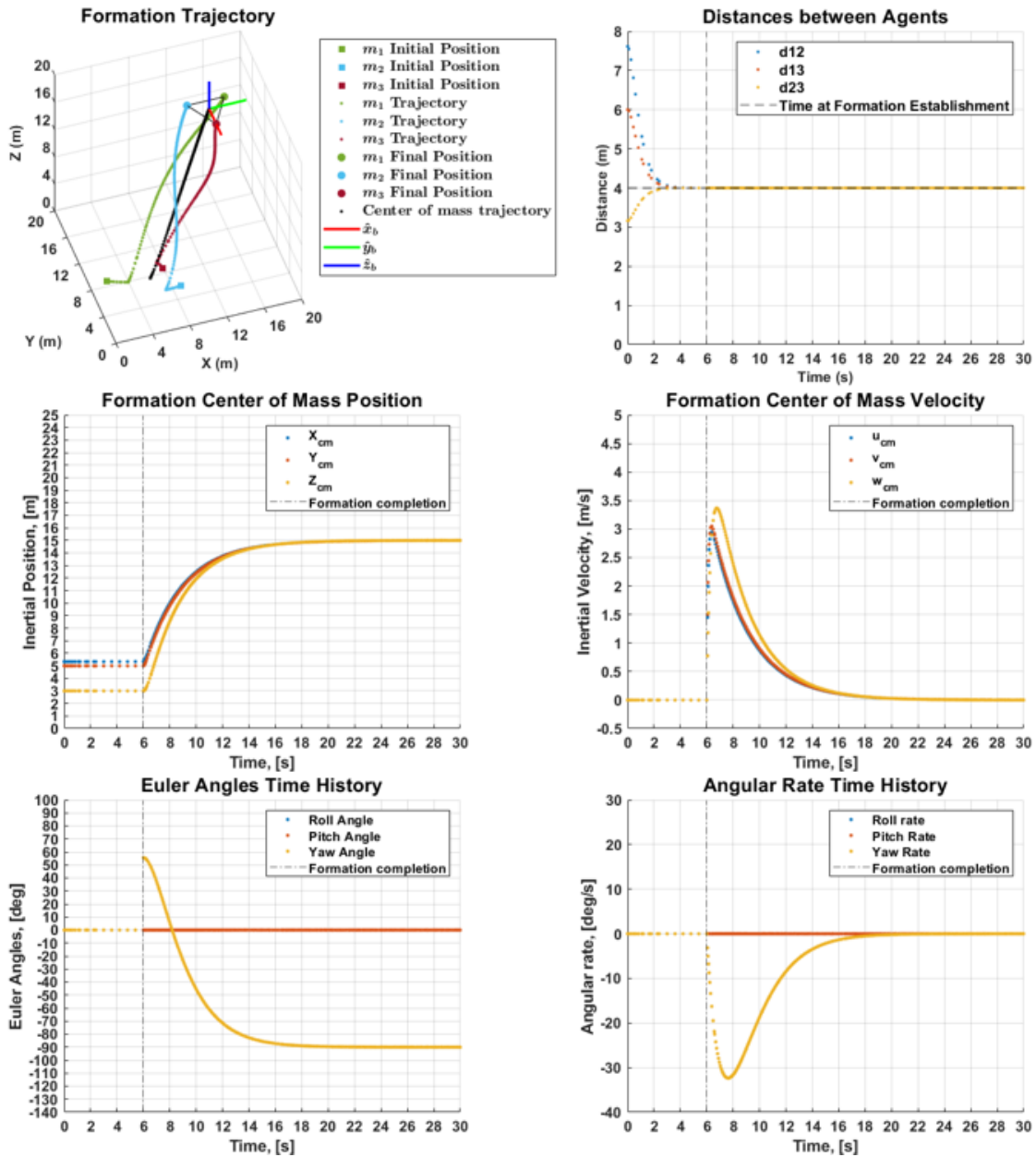


Figure 7.4. Single waypoint simulation results via Local Voting Protocol.

As figure 7.4 shows, formation establishment, formation orientation, and station keeping are all achieved, and all states of the formation also reach their desired values and are stabilized. From these results, this method can be one of the candi-



dates for further consideration. However, depending on the maneuvers that are to be performed, singularity due to Euler angles descriptions may occur.

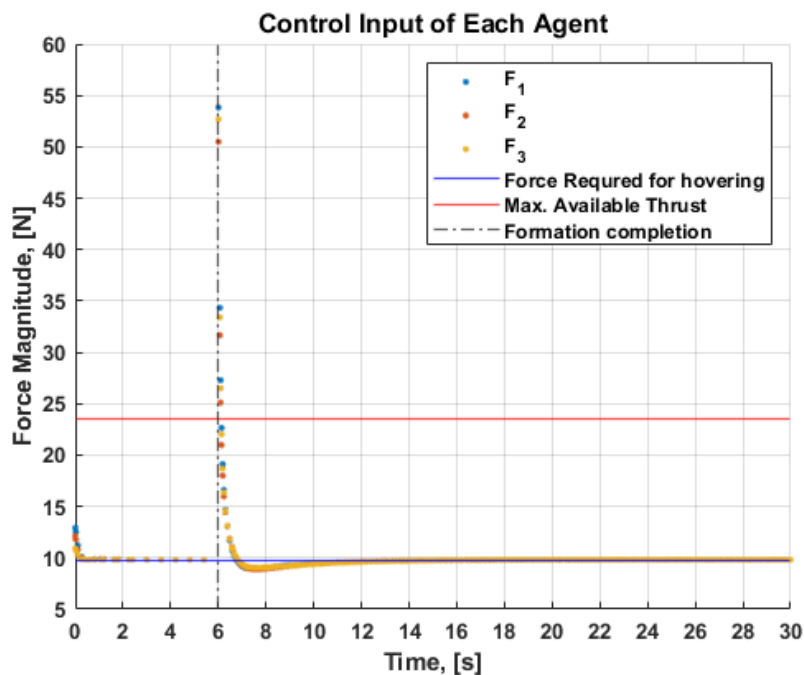


Figure 7.5. Single waypoint simulation result via Local Voting Protocol: Control Inputs.

As you can see from figure 7.5, input force that are higher than the maximum available thrust indicated by [24] are recorded. This problem can be improved by tuning the stiffness and damping coefficients in the equations, and this method can still be one of the candidates for further consideration for simulation in the environment setting due to its ease of implementation; however, its limitation in rotational motion lowers the generality of the framework.

### 7.1.3 Trajectory Tracking

Another supplemental algorithm considered here is the trajectory tracking algorithm. We will first generate a trajectory offline by using an open-loop optimal control method utilizing the Lyapunov equation. Once a set of position and velocity data along with the simulation time are obtained, they are used as reference signals with the local voting protocol. Therefore, reference position and velocity at each time instant are treated as a waypoint. Since the difference between the actual states and the reference states become less compared to the previous method, reduction in control input is expected. Since the final state is fixed at the desired reference signals, it is redundant to include the final-state weighing matrix [23]. Therefore, the solution for minimum input energy is considered. The derivation of an open-loop control method utilizing Lyapunov equation is well explained in [23]. Key components are shown below.

$$\left\{ \begin{array}{l} \dot{x}_{cm} = u_{cm} \\ \dot{y}_{cm} = v_{cm} \\ \dot{z}_{cm} = w_{cm} \\ \dot{u}_{cm} = f_x/m_{cm} \\ \dot{v}_{cm} = f_y/m_{cm} \\ \dot{w}_{cm} = f_z/m_{cm} \end{array} \right. \quad (7.3)$$

Trajectory Tracking with Local Voting Protocol	
Components	Equations
State space	$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}$
Performance Index	$J = \frac{1}{2} \int_0^T \mathbf{U}^T \mathbf{R} \mathbf{U} dt$
Optimal Input	$\mathbf{U}^*(t) = \mathbf{R}^{-1} \mathbf{B}^T e^{\mathbf{A}^T(T-t)} \mathbf{G}(0, T)^{-1} \left\{ \mathbf{X}(T) - e^{\mathbf{A}T} \mathbf{X}(0) \right\}$
Lyapunov Equation	$\mathbf{G}(t_0, T) = \mathbf{P}(T)$ $\dot{\mathbf{P}} = \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T, \mathbf{P}(t_0) = 0$

Table 7.4. Equations used for optimal trajectory generation

By calculating the optimal input, the following optimal trajectory as well as position and velocity data points are obtained.

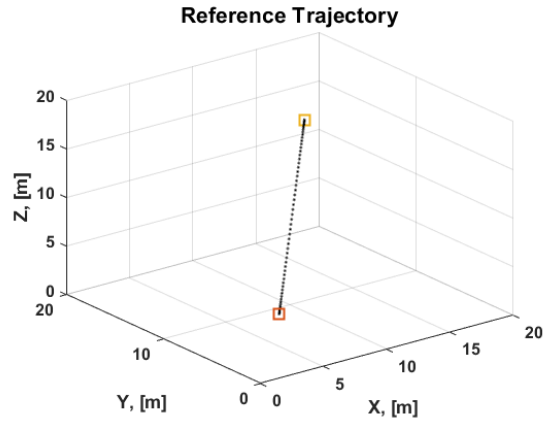


Figure 7.6. Reference trajectory.

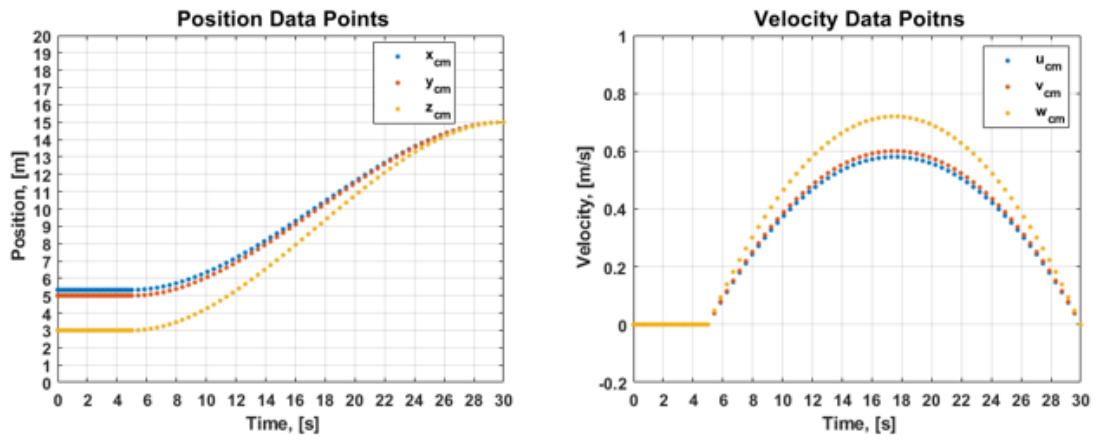


Figure 7.7. Reference position and velocity data.

Now, by supplying these data points as reference signals in the local voting protocols, the following simulation results were obtained.

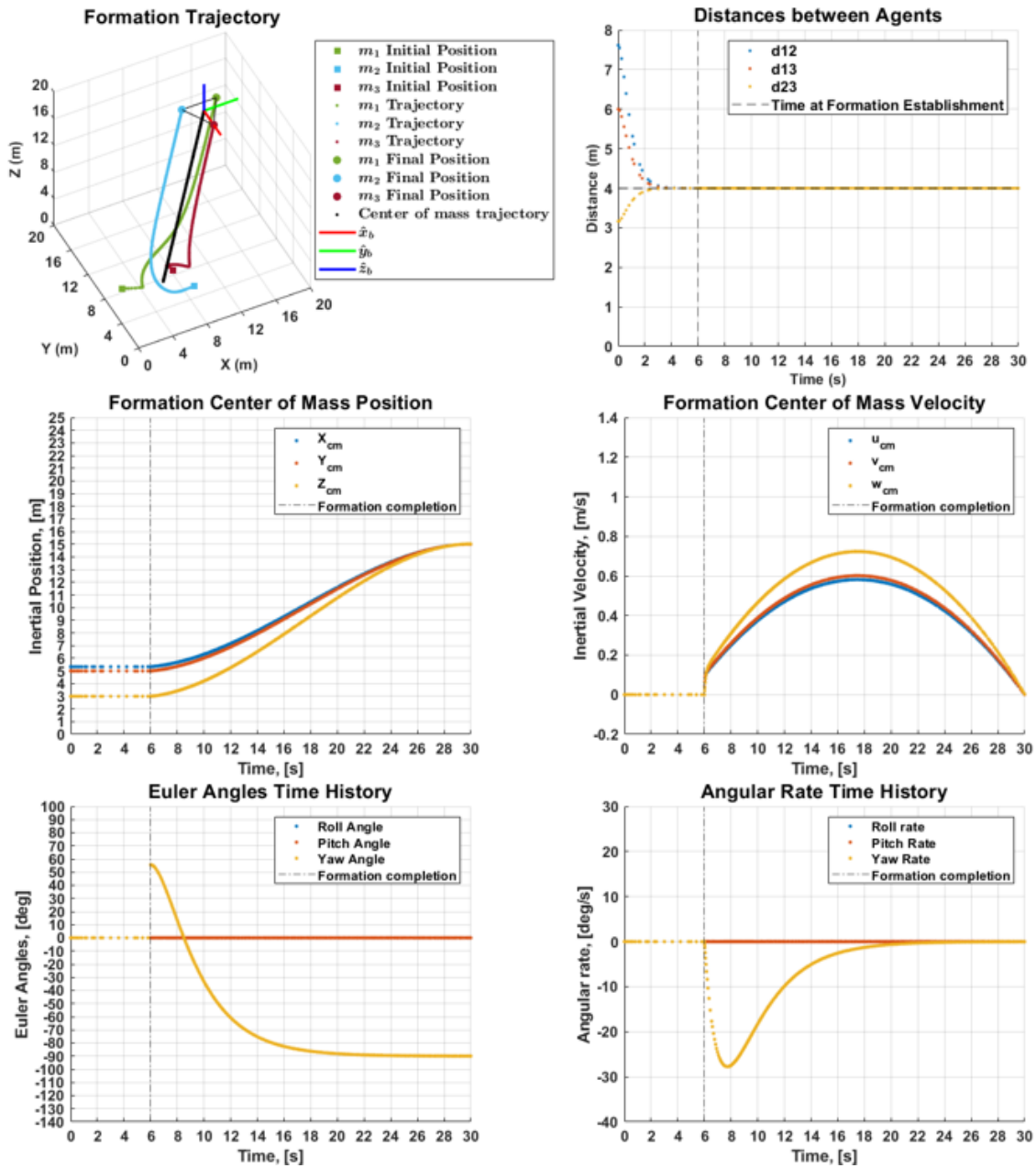


Figure 7.8. Single waypoint simulation results via trajectory tracking.

As figure 7.8 shows, formation establishment, formation maintenance, orientation as well as station-keeping are accomplished with this method as well. From

the trajectory plot, a clear difference in formation behavior is observed compared to previous methods. Moreover, from the comparisons between the simulation results and figure 7.7, both position and velocity signals are well followed.

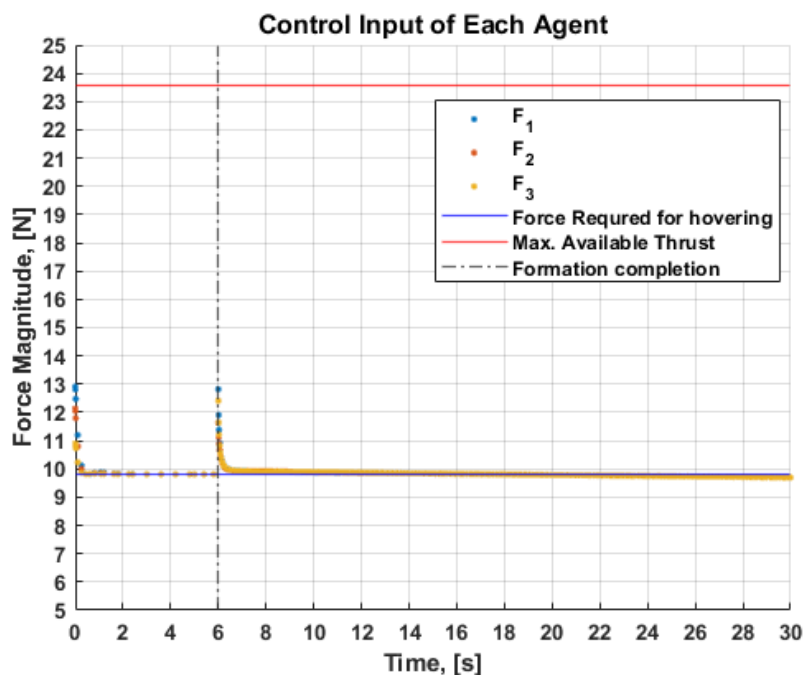


Figure 7.9. Single way-point simulation results for trajectory tracking: control inputs of each agent.

From a comparison between figures 7.5 and 7.9, the control inputs on each agent are reduced drastically as expected. While this method can also be considered as a candidate for further consideration, limitations in the rotational motion still exist. From the simulation results in this section, let us use LQR for further consideration to avoid loss of generality in this framework. Now, let us see if multiple way-point motion is possible in the environment example shown in figure 7.1.

## 7.2 Multiple Waypoint Simulations

In this section, we will synthesize the work done in the previous section to the environment shown in figure 7.1. To achieve this, we will consider a Linear Quadratic Regulator (LQR). Environment descriptions and the mission plan are shown below.

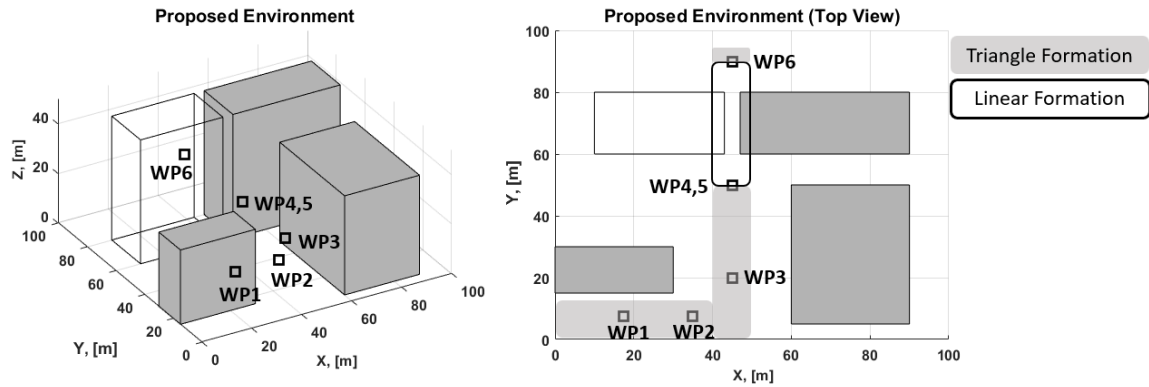


Figure 7.10. Environment descriptions and mission details.

The following way points and their conditions are specified for this environmental setup.

Waypoints Specifications and Descriptions	
Waypoint	Specifications
Waypoint 1	$\mathbf{r}_{cm} = [17.5, 7.5, 20]^T m, \mathbf{v}_{cm} = [0, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 0]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 2	$\mathbf{r}_{cm} = [35, 7.5, 20]^T m, \mathbf{v}_{cm} = [2, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 0]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 3	$\mathbf{r}_{cm} = [45, 20, 20]^T m, \mathbf{v}_{cm} = [0, 2, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 0]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 4	$\mathbf{r}_{cm} = [45, 50, 20]^T m, \mathbf{v}_{cm} = [0, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 0]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 5	$\mathbf{r}_{cm} = [45, 50, 20]^T m, \mathbf{v}_{cm} = [0, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 90]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 6	$\mathbf{r}_{cm} = [45, 90, 20]^T m, \mathbf{v}_{cm} = [0, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 90]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Key Phases	Descriptions
Waypoint 0	Clear the ground
Waypoint 1	Reach certain altitude to gain safe altitude
Waypoint 4	Reconfigure the formation from equilateral triangle to linear
Waypoint 5	Re-orient the formation to the narrow path ahead
Waypoint 6	Retrieve the original equilateral triangle

Table 7.5. Simulation specifications for multiple-waypoint case

Since this mission requires formation reconfiguration, the following Simulink architecture is employed.



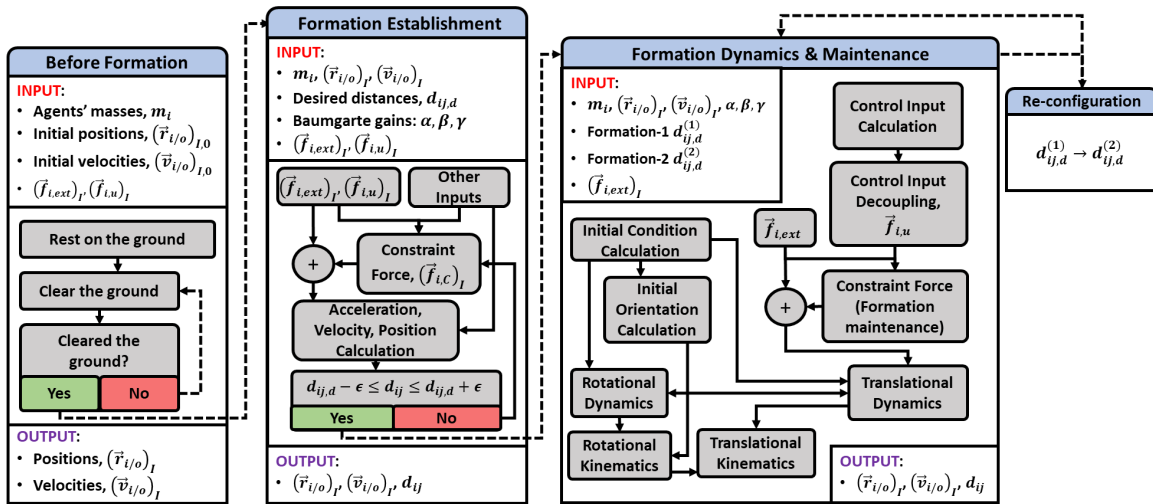


Figure 7.11. Simulink Architecture for Multi-Waypoint flight.

By using LQR for each waypoint interval to calculate the control inputs in the simulink architecture shown in figure 7.11, the following results were obtained.

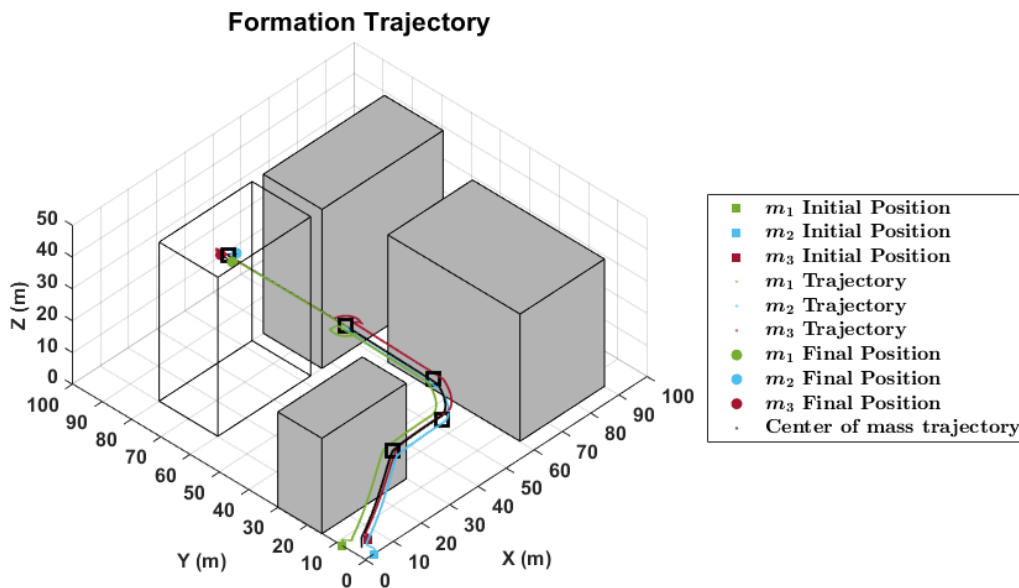


Figure 7.12. Multiple waypoint simulation results via LQR: Formation trajectory.

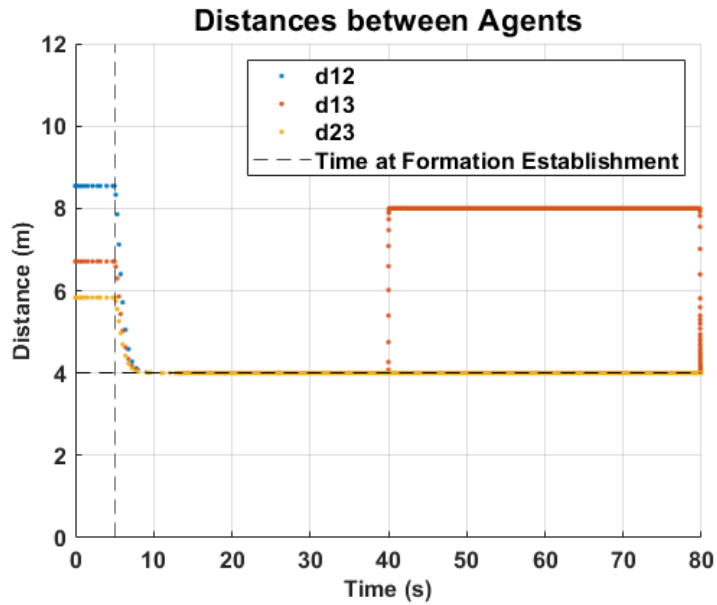


Figure 7.13. Multiple waypoint simulation results via LQR: Distances between agents.

As figures 7.12 and 7.13 show, the formation went through all the way points while maintaining the formation. At way point 4, the formation reconfiguration from a triangular formation to a linear formation was established. As the new formation cleared the narrower path and reached the final destination, the triangular formation was retrieved.

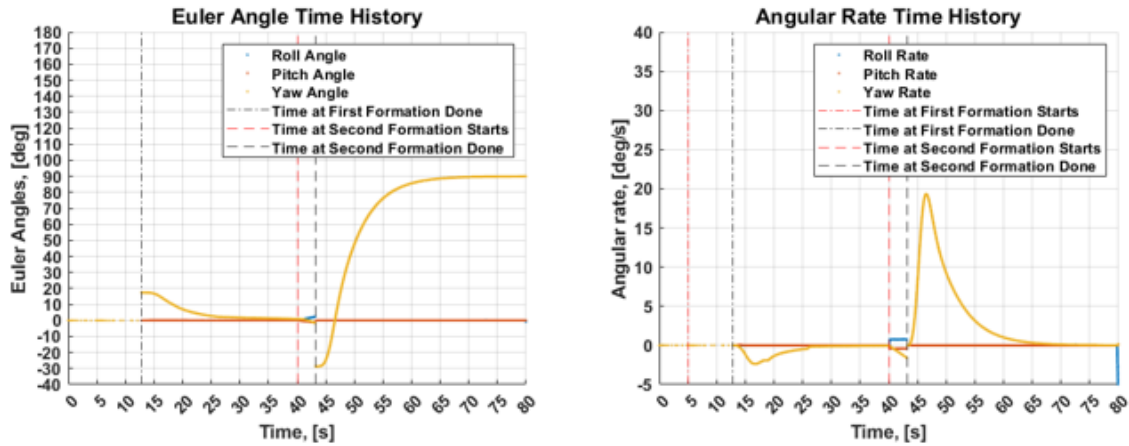


Figure 7.14. Multiple waypoint simulation results via LQR: Formation Euler angles (left), Formation Angular Rates (right).

As it is shown in figure 7.14, desired Euler angles and angular rates were realized both before and after the formation reconfiguration while there are small distortions in them while the formation was shifting due to change in the mass moment of inertia. As the triangular formation was shifted to the linear formation, the heading angle was corrected to  $90^\circ$  so that the formation could clear the narrower path. Now, let us see the control input on each agent.

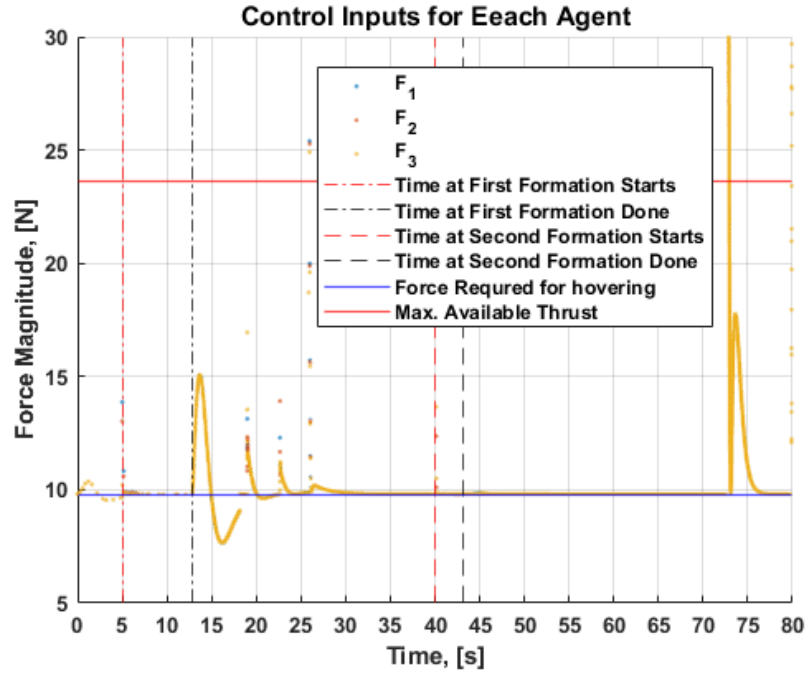


Figure 7.15. Multiple waypoint simulation results via LQR: Agents' Control Inputs.

As figure 7.15 shows, the control inputs stayed below the maximum available thrust most of the simulation period; however, there are periods where the inputs were extremely high. While optimization of control input is not a main objective of this research, it needs to be addressed during future development if implementation of the framework to the physical system is considered. So far, we have shown that multi-waypoint flight is feasible with this framework. Now, let us increase the scale of the formation in terms of the number of agents.

## Chapter 8

### FORMATION WITH HIGHER NUMBER OF AGENTS

We have completed a step-by-step development of a VRB framework and the simulations with three agents. Now, let us increase the number of agents and go through the same process to confirm if the framework is capable of synthesizing formations with a higher number of agents. If the framework is developed correctly, the number of the agents should not affect its viability. In this section, let us consider an eight-agent case.

#### 8.1 Formation Validation

First, let us check if a variety of formations, such as a cubic formation, an octagon formation, and a linear formation can be achieved for an eight-agent case. To form a cubic formation, the following constraint graph is considered. Formation establishment for the octagon and linear formations are shown in Appendix A.

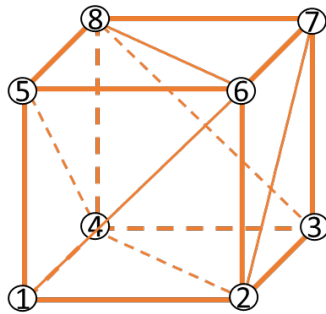


Figure 8.1. Constraint graph of cubic formation.

As figure 8.1 shows, not all possible constraints between agents are imposed. As [11] states, the maximum number of independent constraints for 3D formation is  $3n - 6$ , and any number of constraints over this value will result in an over-defined system. Therefore, eighteen constraints, which is shown in figure 8.1 are carefully chosen for this eight-agent case. The following input specifications are used for this cubic formation simulation.

<b>Input specifications</b>	
<b>Input Description</b>	<b>Values</b>
Number of Agents	$N = 8$
Agent's mass	$m_i = 1kg \forall i = 1, 2, \dots, 8$
Desired distances, $d_{1j,d}$	$[d_{12,d}, d_{14,d}, d_{15,d}, d_{16,d}] = [4, 4, 4, 4\sqrt{2}]m$
Desired distances, $d_{2j,d}$	$[d_{23,d}, d_{24,d}, d_{26,d}, d_{27,d}] = [4, 4\sqrt{2}, 4, 4\sqrt{2}]m$
Desired distances, $d_{3j,d}$	$[d_{34,d}, d_{37,d}, d_{38,d}] = [4, 4, 4\sqrt{2}]m$
Desired distances, $d_{4j,d}$	$[d_{45,d}, d_{48,d}] = [4, 4\sqrt{2}]m$
Desired distances, $d_{5j,d}$	$[d_{56,d}, d_{58,d}] = [4, 4]m$
Desired distances, $d_{6j,d}$	$[d_{67,d}, d_{68,d}] = [4, 4\sqrt{2}]m$
Desired distances, $d_{7j,d}$	$d_{78,d} = 4m$
Initial position	$\mathbf{r}_{1,0} = [2, 1, 1]^T m, \mathbf{r}_{2,0} = [3, 1, 1]^T m, \mathbf{r}_{3,0} = [5, 2, 1]^T m,$ $\mathbf{r}_{4,0} = [1, 5, 1]^T m, \mathbf{r}_{5,0} = [4, 5, 4]^T m, \mathbf{r}_{6,0} = [5, 4, 4]^T m,$ $\mathbf{r}_{7,0} = [7, 8, 4]^T m, \mathbf{r}_{8,0} = [5, 8, 4]^T m$
Initial velocity	$\mathbf{v}_{i,0} = [0, 0, 0]^T m/s \forall i = 1, 2, \dots, 8$
External force	$\mathbf{f}_{i,ext} = [0, 0, -m_i g]^T N$ (gravitational force)
Input force	$\mathbf{f}_{i,u} = [0.1, 0.1, m_i g]^T N$ (constant altitude translation)

Table 8.1. Eight-agent case formation establishment input specifications

As a result, the following simulation results are obtained.

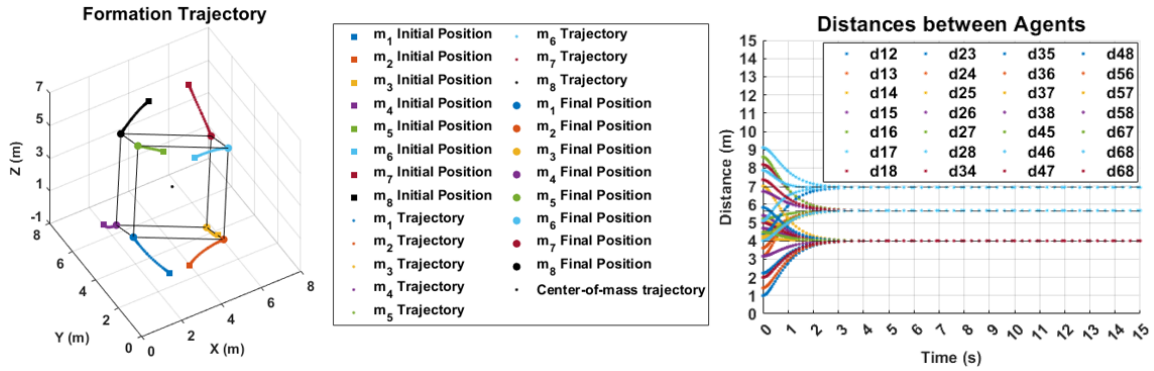


Figure 8.2. Cubic formation simulation results.

As figure 8.2 shows, the desired cubic formation derived from the desired distances between agents were established and stabilized throughout the simulation time.

## 8.2 Body Frame Attachment

To consider a body frame of the VRB, the following body frame is attached.

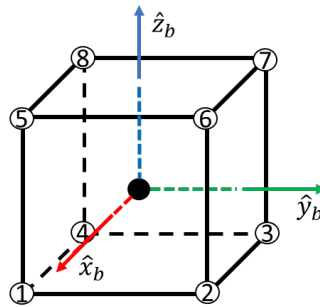


Figure 8.3. Eight-agent formation, body frame orientation.

The  $\hat{x}_B$  frame is in the direction of  $\delta \mathbf{r}_{1,4}$ , the  $\hat{y}_B$  frame is in the direction of  $\delta \mathbf{r}_{2,1}$ , and the  $\hat{z}_B$  can be obtained through a cross product of the  $\hat{x}_B$  and  $\hat{y}_B$  axes. The origin of the body frame is placed at the center of mass of the formation.

### 8.3 Integrated System Validation

The viability of the VRB formation with a higher number of agents has been proven, and the body frame is attached. Thus, let us apply control inputs to it. In the following simulation, a single-waypoint simulation is performed as it was done in the previous case. The following waypoint is considered.

Waypoint Specifications	
Waypoint	Values
C.M. position	$\mathbf{r}_{cm} = [10, 4.25, 4.5]^T m$
C.M. velocity	$\mathbf{v}_{cm} = [0, 0, 0]^T m/s$
Formation orientation	$\boldsymbol{\sigma} = [0, 0, 0]^{\circ T}$
Formation angular rates	$\boldsymbol{\omega}_b = [0, 0, 0]^{\circ T}/s$

Table 8.2. Waypoint specifications for single waypoint case

The same input specifications shown in Table 8.1 are applied to this simulation as well except the control input. By utilizing LQR, which is shown in Table 7.2, the following results are obtained.



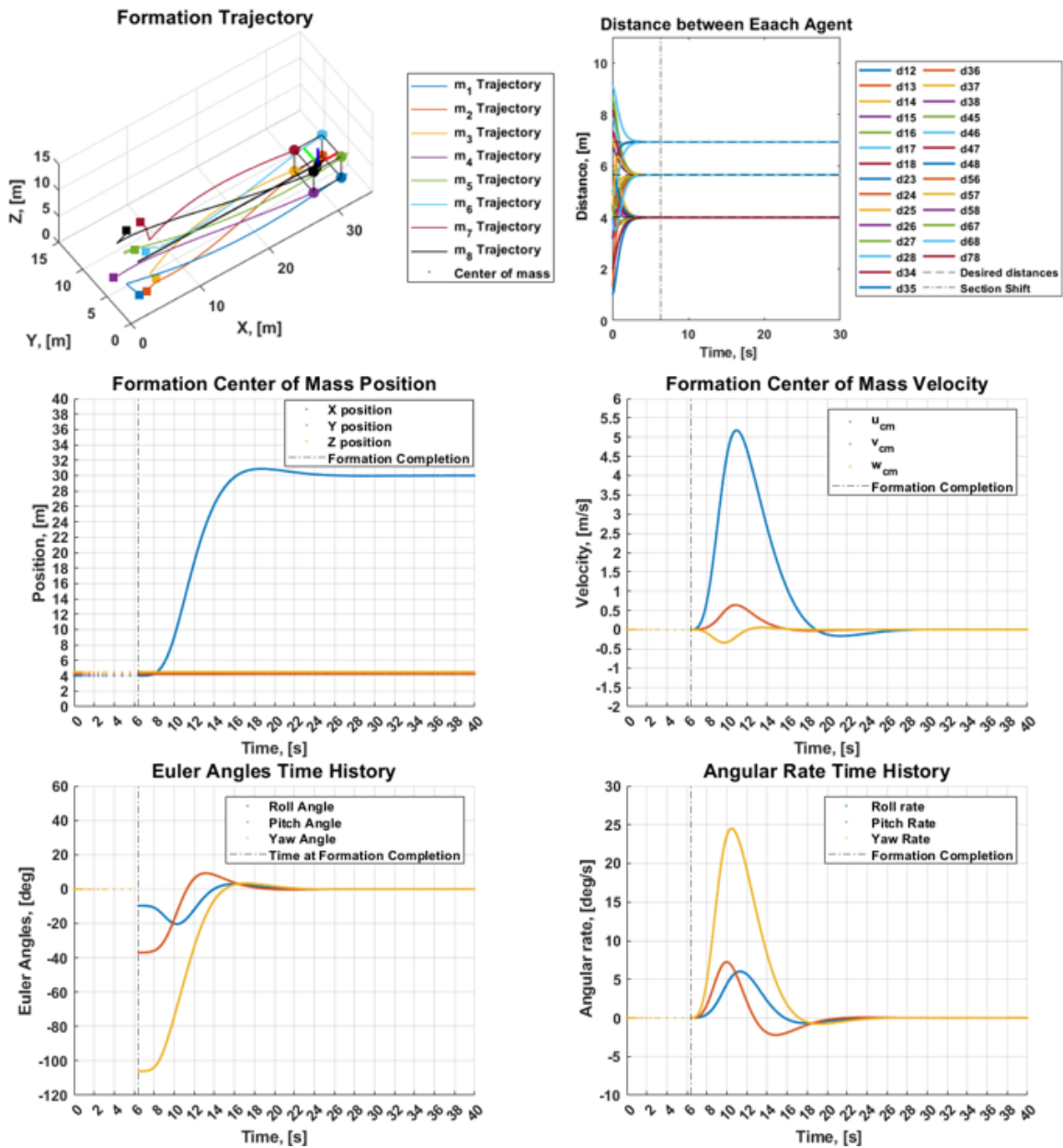


Figure 8.4. Cubic formation single way point simulation results.

As figure 8.4 shows, the desired cubic formation was established by realizing and stabilizing the desired distances between agents. After the formation was established, its position and velocity reached their desired value and were stabilized. Moreover, its

orientation of the formation was initialized, corrected to, and stabilized at the desired values. Most importantly, the formation was maintained throughout the simulation.

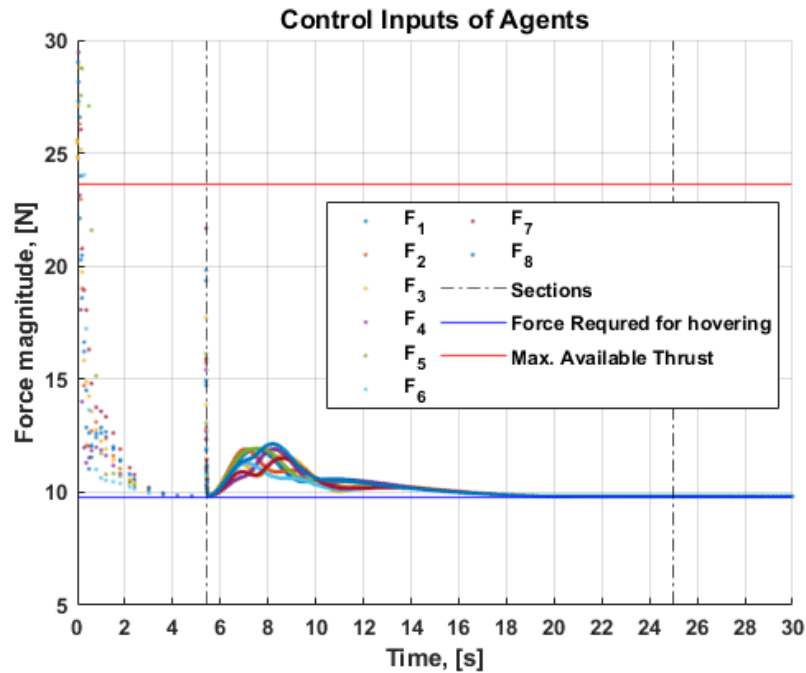


Figure 8.5. Cubic formation single way point simulation: Control inputs on agents.

As figure 8.5 shows, inputs higher than the available thrust are observed at the beginning of the formation establishment section. However, they stay well below the maximum available thrust for most of the simulation period.

#### 8.4 Application Example

We have shown that VRB formation establishment as well as synthesis of linear quadratic method are possible for the eight-agent case in the previous sections. Now, let us see if this framework can be synthesized with the multi-waypoint environment.

The LQR is, again, considered to calculate the control input. The environment and the mission specification are described below.

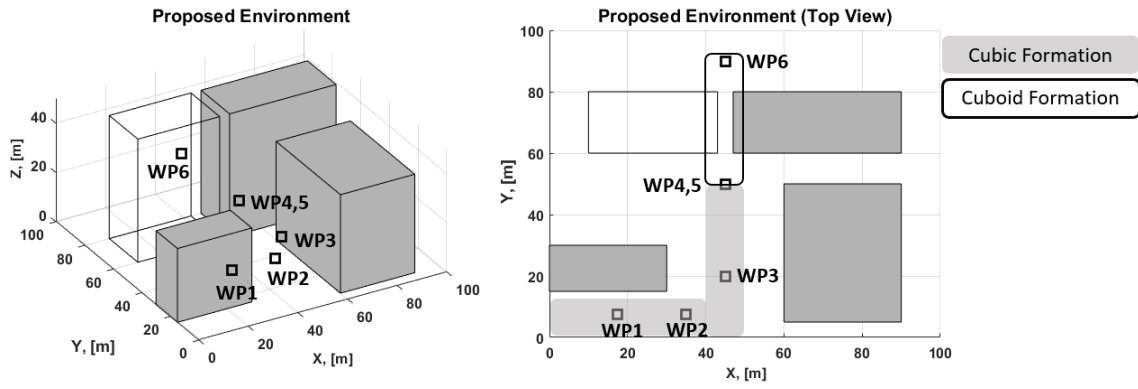


Figure 8.6. Environment and mission specifications.

Since different formation configuration is considered in this section, a couple of updates in the waypoints are made.

Waypoints Specifications and Descriptions	
Waypoint	Specifications
Waypoint 1	$\mathbf{r}_{cm} = [17.5, 7.5, 20]^T m, \mathbf{v}_{cm} = [0, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 0]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 2	$\mathbf{r}_{cm} = [35, 7.5, 20]^T m, \mathbf{v}_{cm} = [2, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 0]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 3	$\mathbf{r}_{cm} = [45, 20, 20]^T m, \mathbf{v}_{cm} = [0, 2, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 0]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 4	$\mathbf{r}_{cm} = [45, 50, 20]^T m, \mathbf{v}_{cm} = [0, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 0]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 5	$\mathbf{r}_{cm} = [45, 50, 20]^T m, \mathbf{v}_{cm} = [0, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 0]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Waypoint 6	$\mathbf{r}_{cm} = [45, 90, 20]^T m, \mathbf{v}_{cm} = [0, 0, 0]^T m/s,$ $\boldsymbol{\sigma} = [0, 0, 90]^{T^\circ}, \boldsymbol{\omega}_b = [0, 0, 0]^{T^\circ}/s$
Key Phases	Descriptions
Waypoint 0	Clear the ground
Waypoint 1	Reach certain altitude to gain safe altitude
Waypoint 4	Reconfigure the formation from cubic to cuboid formation
Waypoint 5	Re-orient the formation to the narrow path ahead
Waypoint 6	Orient the cuboid formation by $90^\circ$

Table 8.3. Simulation specifications for multiple-waypoint case: Eight-agent case

As a result of a simulation, the following outcomes were obtained.

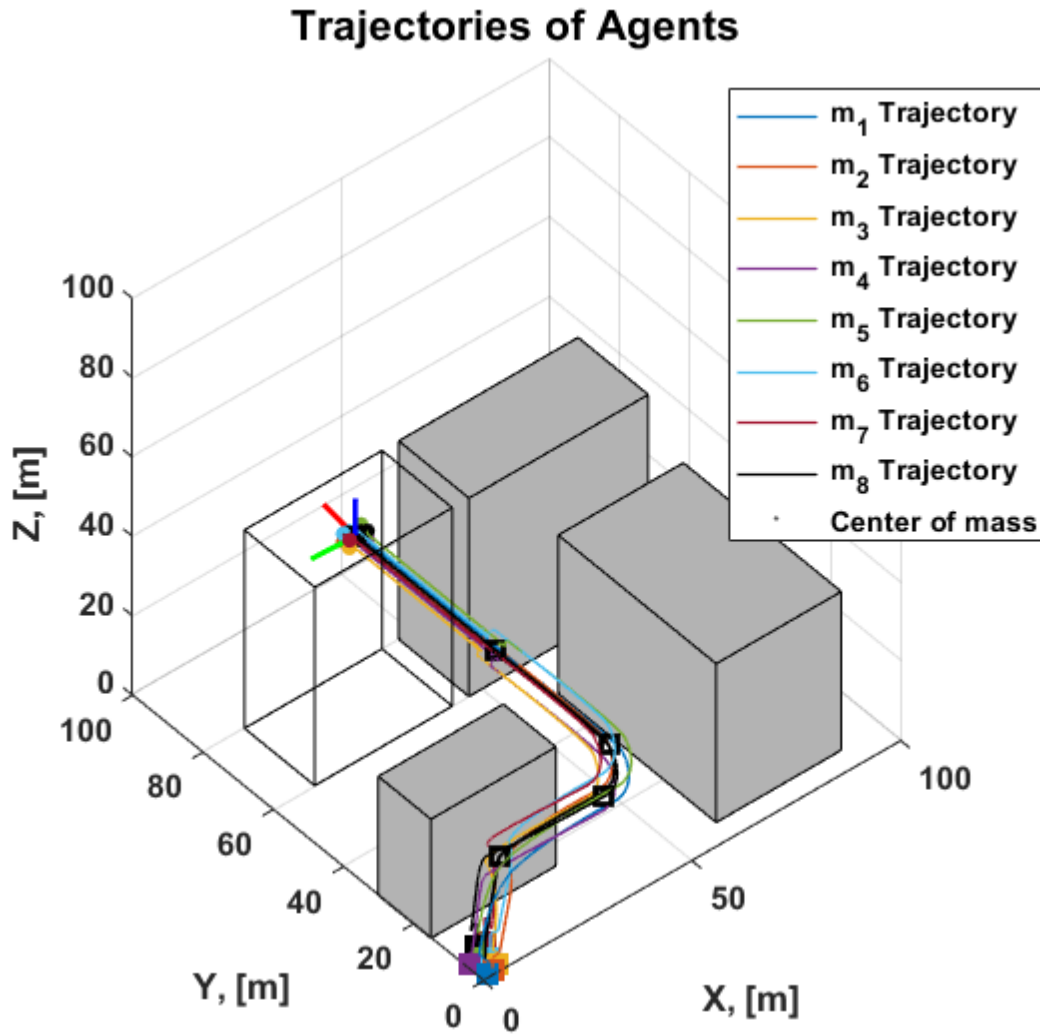


Figure 8.7. Eight-agent VRB formation simulation result.

As figure 8.7 shows, the cubic formation went through or near each waypoint. At the fourth waypoint, the cubic formation changed its shape to a cuboid formation so that it could clear the narrower environment. As the formation reached the destination, it was rotated by  $90^\circ$  about the  $\hat{z}_B$  axis.

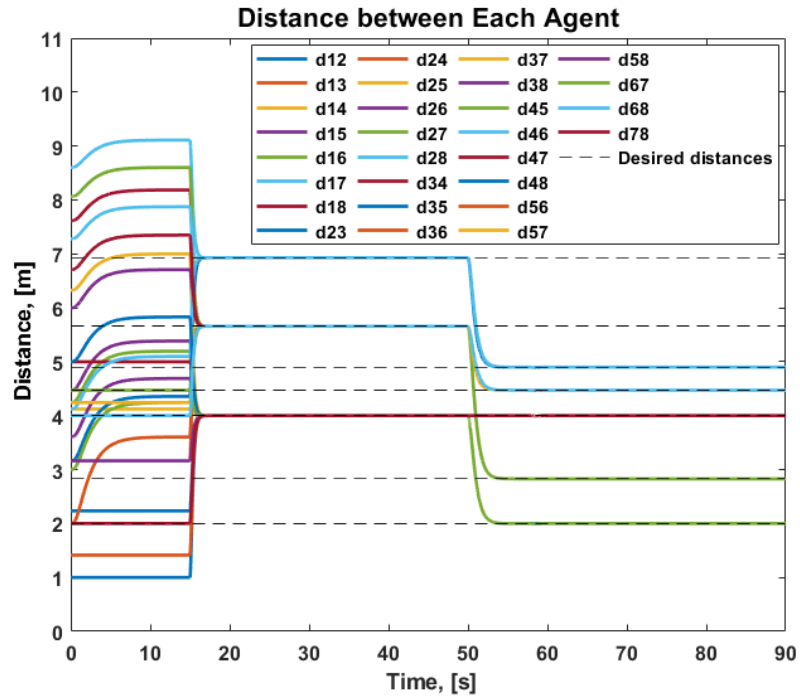


Figure 8.8. Eight-agent VRB formation simulation result: Distances between agents.

As it can be seen from figure 8.8, after the formation algorithm was executed, the cubic rigid formation was quickly formed and maintained until the reconfiguration to the cuboid formation was commanded. After another formation configuration was achieved, the formation was maintained for the rest of the simulation.

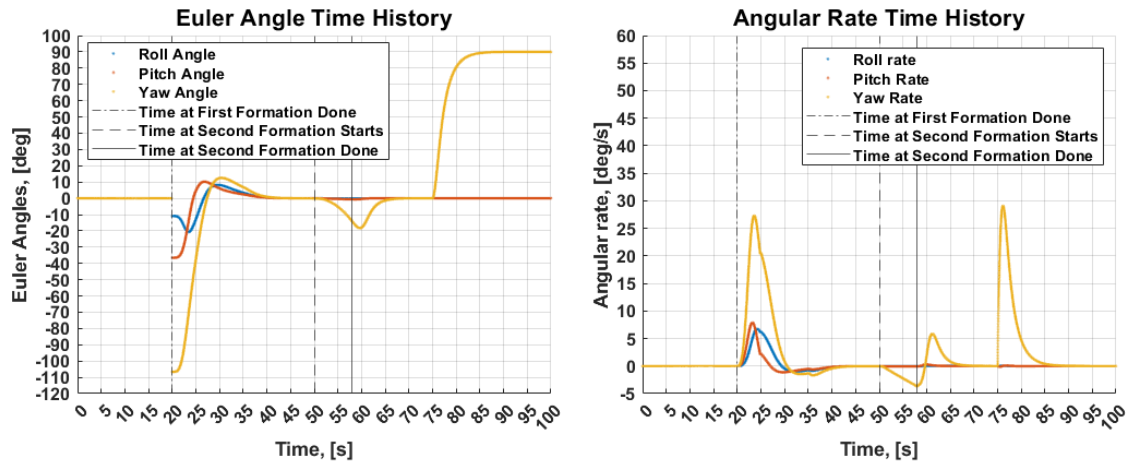


Figure 8.9. Eight-agent VRB formation simulation results: Euler angles and angular rates.

As figure 8.9 shows, after the formation establishment, the desired angular rates and the Euler angles are achieved both before and after formation reconfiguration. At the end of the mission, the formation was rotated by  $90^\circ$  about the  $\hat{z}_B$  axis.

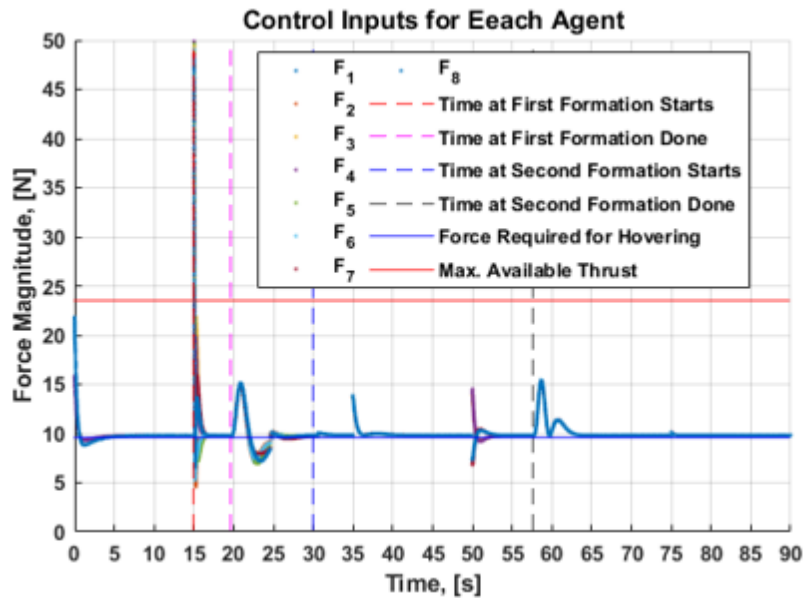


Figure 8.10. Eight-agent VRB formation simulation results: Control Input.

As figure 8.10 shows, control inputs higher than the maximum available thrust were recorded at the beginning of the trajectory. However, before and after that time instant, the control input stayed below the maximum available value. From these results, while investigation in control input optimization needs to be performed, it can be concluded that formation establishment, maintenance, reorientation, reconfiguration, and station-keeping of the higher number of agents as well as LQR synthesis are highly viable.



## Chapter 9

### SUMMARY, CONCLUSION, AND FUTURE WORK

#### 9.1 Summary

This research summarizes the thorough mathematical modeling, control law development, and simulations of the virtual rigid structure formations which are inspired by the characteristics of rigid bodies. By considering autonomous agents as the fixed nodes on a rigid body, the trajectory of each agent on it is automatically determined based on the behavior and motion of the rigid body. To be capable of simulating the motion of each particle, Newton's and Euler's equation are utilized to derive 6-DOF governing equations of motion of a multiagent system. The quaternion kinematic equation is considered to eliminate singularities resulting from the description using the Euler angles and to ensure all-attitude capability. Upon development of the distance-based virtual rigid body formation, the concept of the constraint force is employed, and d'Alembert's principle of virtual work and constraint sensitivities (Lagrange multiplier) are synthesized. a PID-like control derived from the Baumgarte stabilization method is also synthesized with in the constraint force to achieve stable constraint force. Since the governing equations of motion of a multiagent system consider a body frame associated with it, attachment of a body frame to the virtual rigid body is performed from geometric characteristics of a formation. Moreover, input decoupling method is introduced to distribute the inputs precisely and properly to each agent so that the collective motion of agents generates the desired behavior of the VRB formation. After necessary components are developed, they are synthesized in the Simulink properly, and the framework is simulated under multiple situations.

Beginning with simple inputs, multiple maneuvers are tested. As the framework is validated with the simple control inputs, the system's feasibility under the multiple input calculation methods, such as linear quadratic method, local voting protocol, trajectory tracking algorithm, are tested via the single waypoint situation. As a result, LQR was selected as further consideration, and the framework was simulated under multiple waypoint environment involving multiple key scenario. At the end, scale of the framework was increased from three agents to eight agents, and the same step-by-step simulations were carried out to confirm the viability of the framework under the larger scale.

## 9.2 Conclusion

We completed the thorough mathematical modeling and simulation of a virtual rigid body formation that is capable of formation establishment, formation maintenance, reorientation, reconfiguration, and station-keeping for arbitrary number of agents. As a first step of the mathematical modeling, Newton-Euler equations are employed to develop the 6-DOF governing equations of motion of a multiagent system. The concept of constraint force is utilized to develop rigidity in the formation. The constraint force is established by synthesizing d'Alembert's principle of virtual work, constraint sensitivities (Lagrange multiplier), and PID-like constraint stabilization using Baumgarte stabilization technique. After developing other necessary components such as the formation body frame attachment and input decoupling method, all components are synthesized in the Simulink to complete the framework. As a first step of framework feasibility verification, the formation establishment including the formation reconfiguration were tested. As a result, the framework capabilities for formation establishment as well as formation reconfiguration and station-keeping were verified. As the next step, torque inputs were introduced to the system to test

the framework under the rotational motions. During the simulations, steady-state error in the formation establishment was observed. This problem was resolved by tuning the gains introduced in the PID-like control law derived from the Baumgarte stabilization technique. Moreover, thanks to the Quaternions, kinematic singularities resulting from usage of Euler angles were eliminated. As a result, besides the formation establishment and reconfiguration, the formation maintenance and the formation orientation were verified from these simulations. Having verified the integrated framework, each of linear quadratic method, local voting protocol, and trajectory tracking methods were synthesized to test the framework within an application-like setup. Starting off single-waypoint scenario, all methods were found to be compatible to the framework and capable of completing the predefined mission. Motivated by the results from this simulation results, the simulation within multiple-waypoint environment, which includes several key occasions, was carried out with employment of LQR. As a result, all formation establishment, maintenance, orientation, reconfiguration, and station-keeping were successful. However, the input on each agent exceeded the maximum available thrust indicated by a former ASL researcher; thus, further development may be required when this framework is implemented into a physical system. Finally, the scale of this framework was increased from three agents to eight agents to check the viability of the framework under the larger scale. After repeating the same process taken in the three agent case, it was shown that formation establishment, maintenance, orientation, reconfiguration, and station-keeping were all viable with this framework.

### 9.3 Future Work

As future work of this research, we consider improvements in the framework as well as implementation of it in physical systems such as UAVs.

As improvements of the framework, we could 1) synthesize the collision avoidance prior to the formation establishment, and 2) develop swarm of multiple VRB formations. As a VRB formation is established, collision free trajectories for all agents are guaranteed; however, collisions before the formation establishment could still occur. Thus, implementation of collision avoidance algorithm is the necessary improvement. Moreover, by synthesizing multiple VRB formations within the concept of smooth streamlines or potential field method, collision-free swarm of VRB formations, which could widen the applications of the VRB formations, can be developed.

When we consider implementation of the framework in physical systems such as UAVs, it may be ideal to consider a decentralized method, such that each quadcopter in the swarm sees nodes on a VRB as virtual leaders to be tracked. By doing so, communication burdens between UAVs can be minimized and the system can be robust at the event of vehicle losses.

## APPENDIX A

### Additional Formation Cases

## A.1 Two-Agent Case

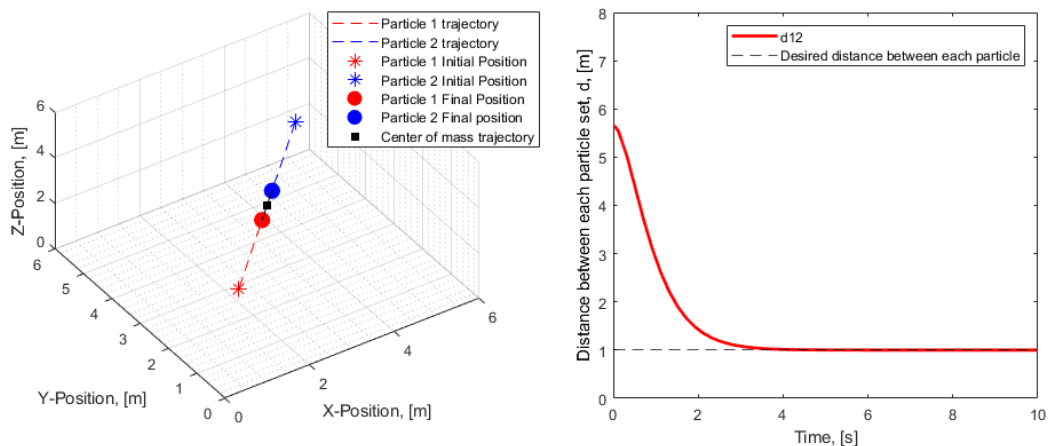


Figure A.1. Two agent formation result (Left): agent trajectories, (Right): Time history of distance between the two agents..

## A.2 Four-Agent Case

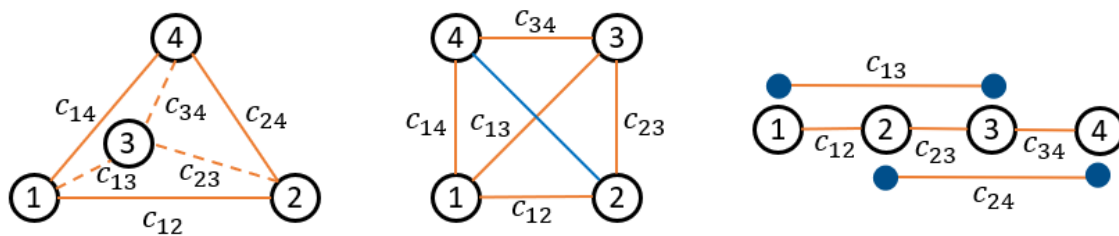


Figure A.2. Four-agent cases constraint graph. (Left): Equilateral tripod formation, (Middle): Square formation, (Right): Linear formation. Orange lines - applied constraints..

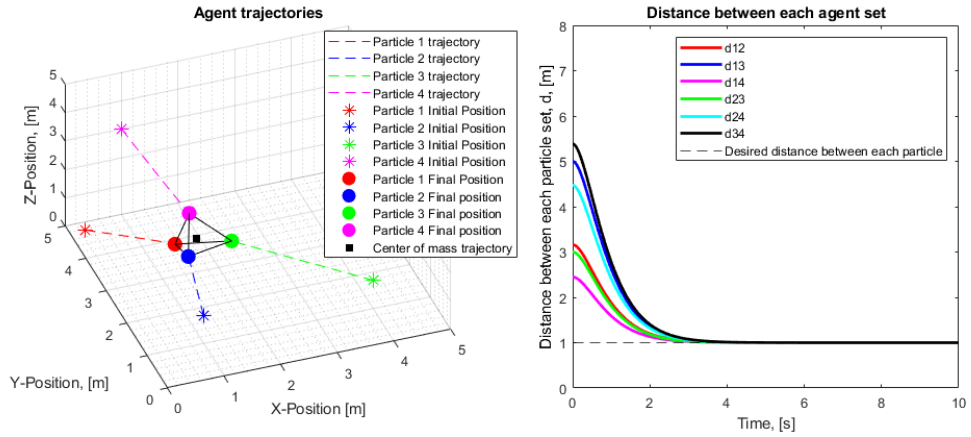


Figure A.3. Four-agent equilateral tripod formation simulation results.

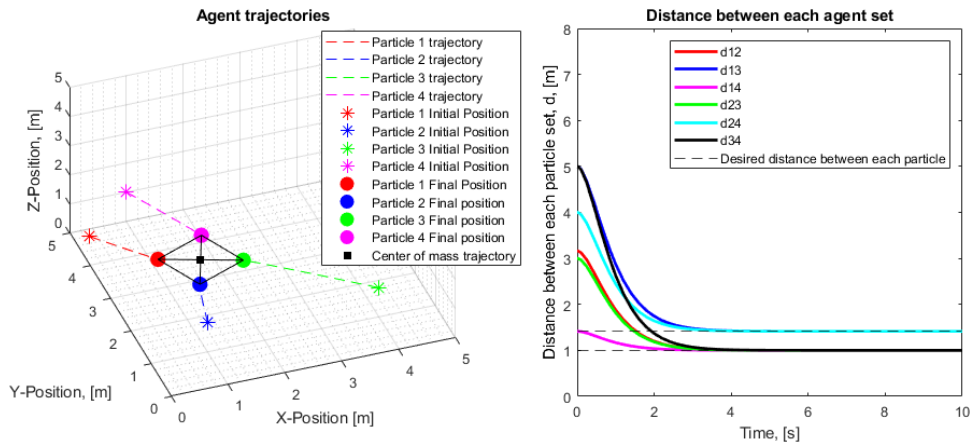


Figure A.4. Four-agent square formation simulation results.

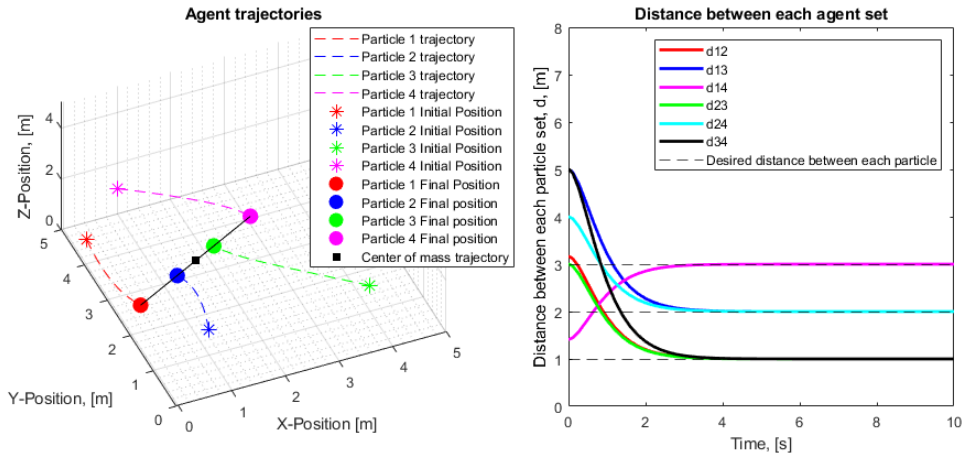


Figure A.5. Four-agent linear formation simulation results.

### A.3 Five-Agent Case

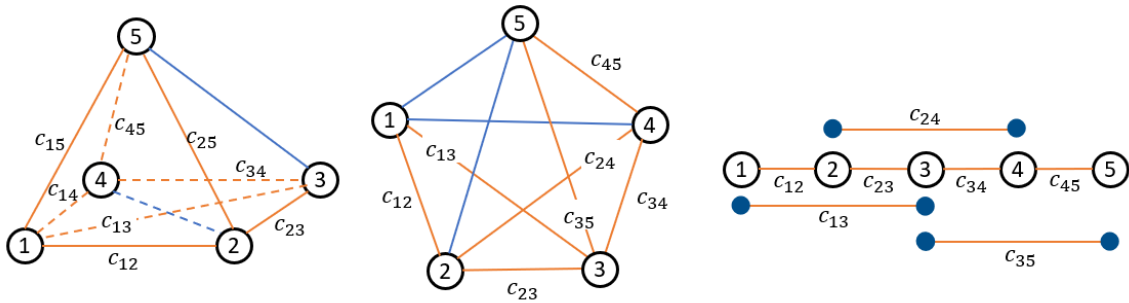


Figure A.6. Five-agent case constraint graph. (Left): Equilateral pyramid formation, (Middle): Pentagon formation, (Right): Linear formation. Orange lines - applied constraints..



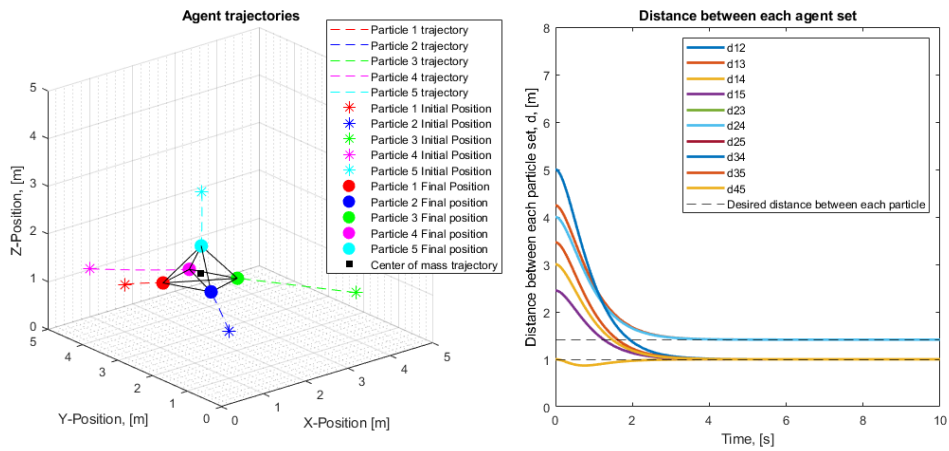


Figure A.7. Equilateral pyramid formation simulation results.

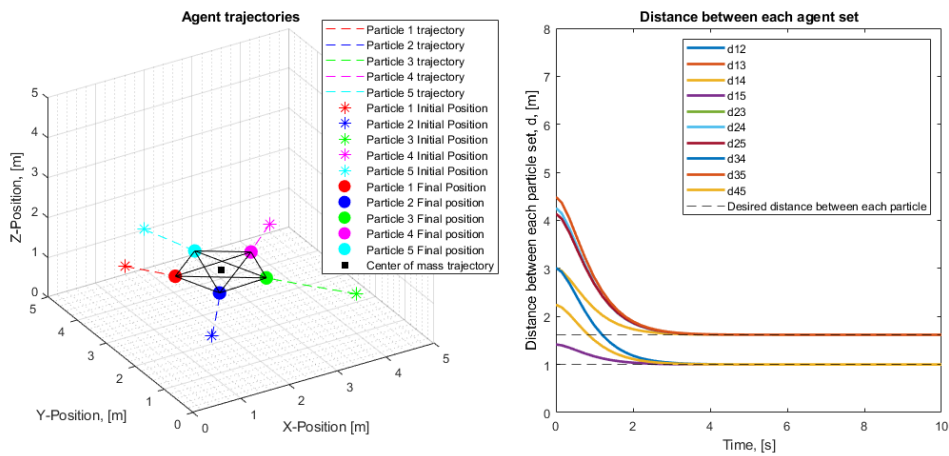


Figure A.8. Pentagon formation simulation results.

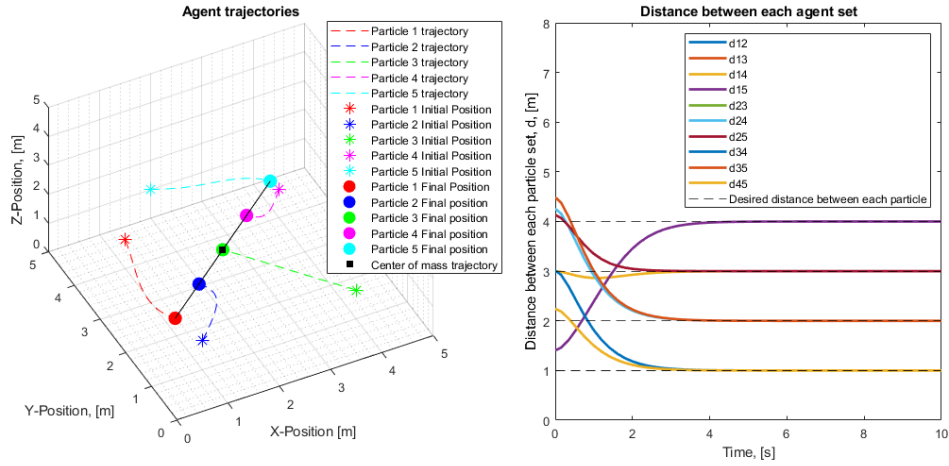


Figure A.9. Linear formation simulation results.

#### A.4 Six-Agent Case

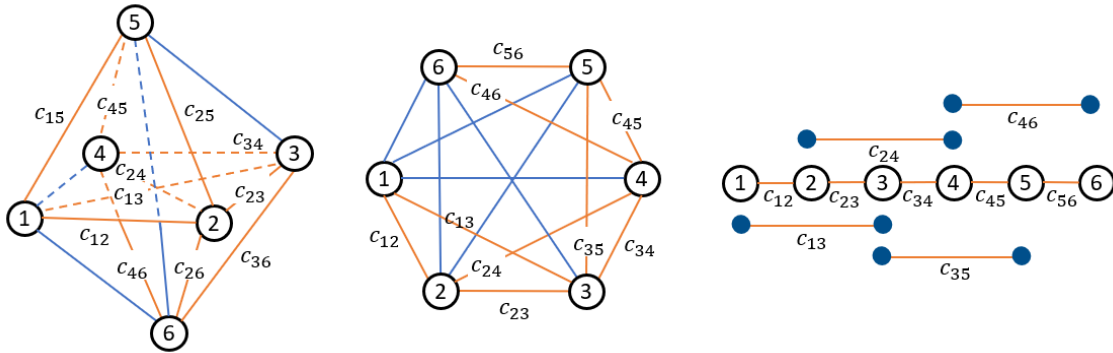


Figure A.10. Six-agent case constraint graph. (Left): Equilateral Dual-pyramid formation, (Middle): Hexagon formation, (Right): Linear formation. Orange lines - applied constraints..

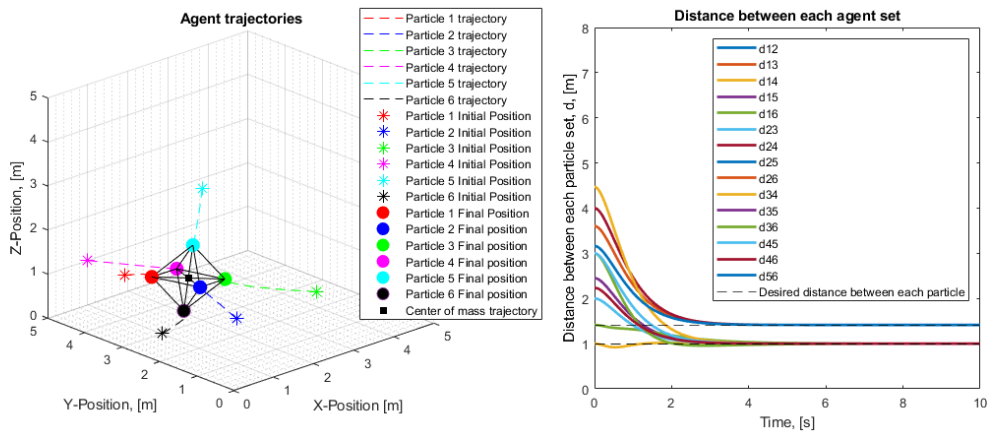


Figure A.11. Equilateral dual-pyramid formation simulation results.

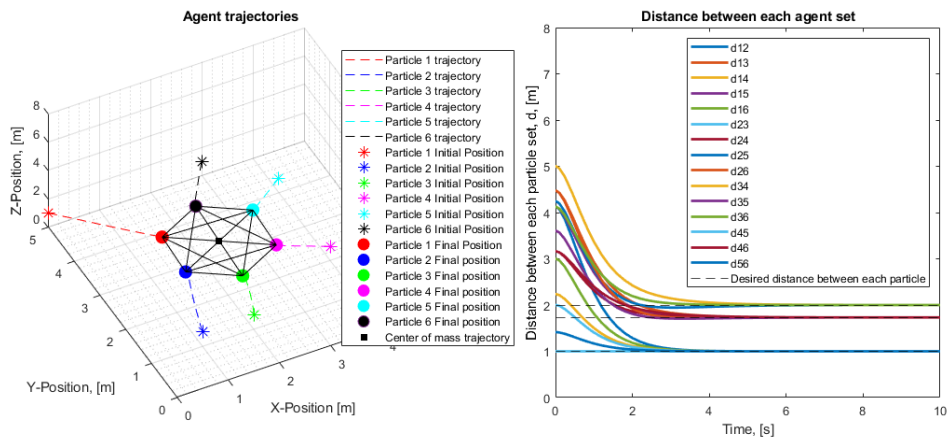


Figure A.12. Hexagon formation simulation results.

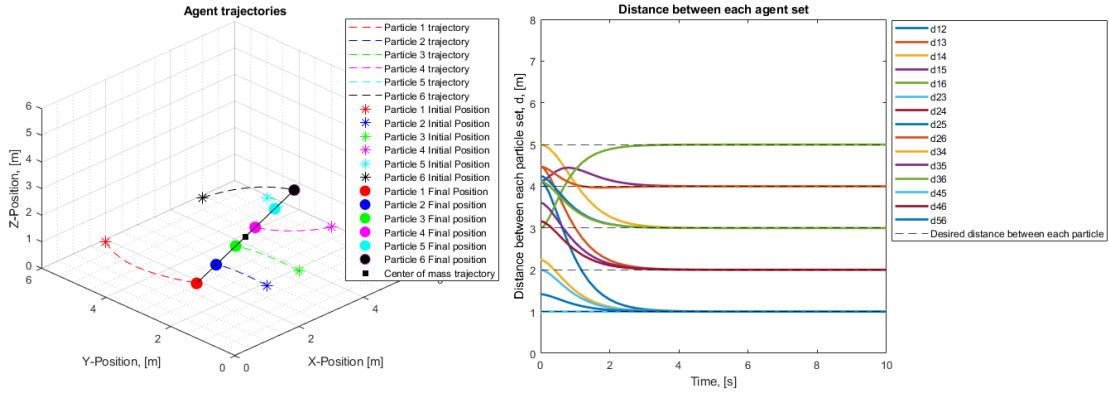


Figure A.13. Linear formation simulation results.

### A.5 Eight-Agent Case

Besides the cubic formation shown in the previous section, cuboid and linear formation are also simulated to show variety of formations. For both formations, the same constraint sequence as the cubic formation is utilized.

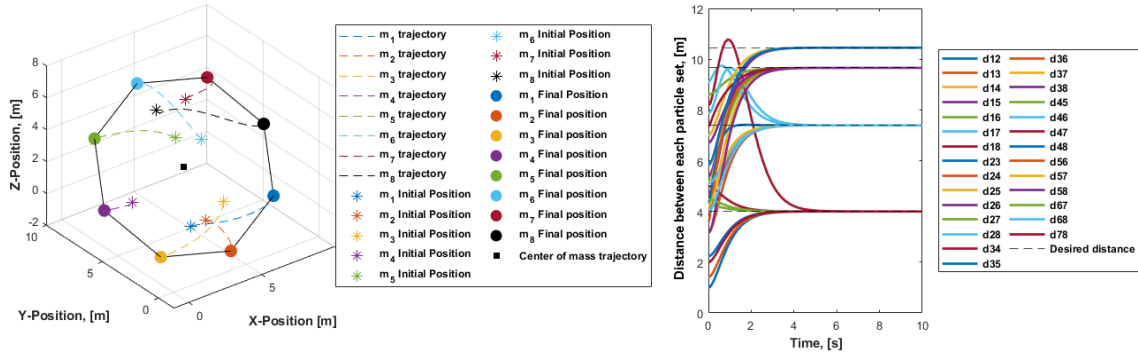


Figure A.14. Octagon formation simulation results.

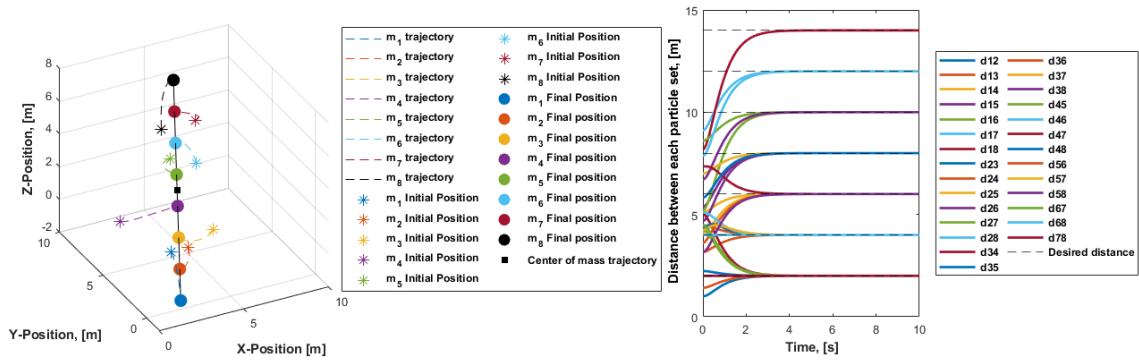


Figure A.15. Linear formation simulation results.

## References

- [1] M. Mesbahi, and F. Y. Hadaegh, "Formation flying control of multiple spacecraft - Graph theoretic properties and switching schemes", *AIAA, Guidance, Navigation, and Control Conference and Exhibit, Portland, OR, USA*, August, 1999, doi: <https://doi.org/10.2514/6.1999-4268>
- [2] I.-A.F.Ihle, J. Jouffroy and T. I. Fossen, "Formation Control of Marine Craft using Constraint Functions", *Proceedings of MTS/IEEE*, Washington, DC, USA, 2005, doi: <https://doi.org/10.1109/OCEANS.2005.1639888>
- [3] D. Hummel, "Formation Flight As Energy-Saving Mechanism", *Israel journal of Zoology*, vol. 41, pp. 261-278, March, 1995, doi: <https://doi.org/10.1080/00212210.1995.10688799>
- [4] V. Chepizhenko, and O. Aliakin, "Analysis of Unmanned Aerial Vehicle Formation Flight Control Method", *Proceedings of the National Aviation University*, pp.22-27, Kyiv, Ukraine, 2015, doi: <https://doi.org/10.18372/2306-1472.64.8929>
- [5] Y. Zou, P.R. Pagilla, and R.T. Ratliff, "Distributed Formation control of multiple aircraft using constraint forces", *American Control Conference*, pp.644-649, Seattle, WA, USA, June, 11-13, 2008, doi: <https://doi.org/10.1109/ACC.2008.4586565>
- [6] R. Cajo, T.M. Thi, C. Copot, D. Plaza, R.D. Keyser, and C. Ionescu, "Multiple UAVs Formation for Emergency Equipment and Medicines Delivery Based on Optimal Fractional Order Controllers", *IEEE, International Conference on Sys-*

- tems, Man and Cybernetics (SMC)*, pp.318-323 Bari, Italy, October 6-9, 2019, doi: <https://doi.org/10.1109/SMC.2019.8914431>
- [7] Bhatia, D., “Constrained Dynamics Approach For Motion Synchronization And Consensus”, Master’s degree thesis, *The University of Texas at Arlington*, 2010
- [8] I.-A.F. Ihle, J. Jouffroy and T.I. Fossen, “Formation Control of Marine Surface Craft using Lagrange Multipliers”, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pp.752-758, Seville, Spain, December 12-15, 2005, doi: <https://doi.org/10.1109/CDC.2005.1582247>
- [9] B. Yun, B.M. Chen, K.Y. Lum, and T.H. Lee, “A Leader-Follower Formation Flight Control Scheme for UAV Helicopters”, *Proceedings of the IEEE International Conference on Automation and Logistics*, pp.39-44, Qingdao, China, September, 2008, doi: <https://doi.org/10.1109/ICAL.2008.4636116>.
- [10] D. Zhou, and M. Schwager, “Virtual Rigid Bodies for Agile Coordination of Quadrotor Swarms and Human-Swarm Teleoperation”, *Boston University and Stanford University*, 2015
- [11] Y. Zou, P.R. Pagilla, and E. Misawa, “Formation of a Group of Vehicles With Full Information Using Constraint Forces”, *ASME Journal of Dynamics Systems, Measurement, and Control*, pp. 654-661, May, 2006, doi: <https://doi.org/10.1115/1.2767659>
- [12] D. Bhatia, and K. Subbarao, “Attitude Formation Control of Multiple Vehicle Subject to Communication Constraints”, *American Institute of Aeronautics and Astronautics Infotech Conference*, St. Louis, Missouri, March, 29 - 31, 2011, doi: <https://doi.org/10.2514/6.2011-1409>
- [13] Y. Zou, and P.R. Pagilla, “Distributed Constraint Force Approach for Coordination of Multiple Mobile Robots”, *Journal of Intelligent and Robotic*

- Systems*, vol 56, pp. 5-21, *Springer*, 2009, doi: <https://doi.org/10.1007/s10846-009-9314-1>
- [14] H. Schaub, and J.L. Junkins, *Analytical Mechanics of Space Systems*, AIAA, 4th edition, Reston, VA, 2019
- [15] F.L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative Control of Multi-Agent Systems - Optimal and Adaptive Design Approaches*, *Springer*, 2014
- [16] Z. Bilgin, M. Bronz, and I. Yavrucuk, “Panel Method Based Guidance for Fixed Wing Micro Aerial Vehicles”, *13th International Micro Air Vehicle Conference*, Delft, Netherlands, September 12-16, 2022, url: <https://www.imavs.org/papers/2022/2.pdf>
- [17] Z. Bilgin, M. Bronz, and I. Yavrucuk, “Experimental Evaluation of Robustness of Panel-Method-Based Path Planning for Urban Air Mobility”, *AIAA AVIATION 2022 Forum*, Chicago, France, June, 2022, doi: <https://doi.org/10.2514/6.2022-3509>
- [18] C. Valeriy, and S. Ihor, “Method of the Multi-UAV Formation Flight Control”, *ICTERI 2018 - 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*, vol 1, pp.167-178, 2018, url: <https://ceur-ws.org/Vol-2105/10000167.pdf>
- [19] J. Sun, J. Tang, and S. Lao, “Collision Avoidance for cooperative UAVs With Optimized Artificial Potential Field Algorithm”, *IEEE Access*, vol 5, pp. 18382 - 18390, September 27, 2017, doi: <https://doi.org/10.1109/ACCESS.2017.2746752>
- [20] B.L. Stevens, F.L. Lewis, and E.N. Johnson, *Aircraft Control and Simulation*, Wiley, 3rd edition, Hoboken, NJ, 2026



- [21] T. Eren, “Rigid Formation of Autonomous Agents”, Ph.D. thesis, Yale University, 2004
- [22] J. Baumgarte, “Stabilization of Constraints and Integrals of Motion in Dynamical Systems”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 1, Vol. 1, issue 1, pp. 1-16, 1972, doi: [https://doi.org/10.1016/0045-7825\(72\)90018-7](https://doi.org/10.1016/0045-7825(72)90018-7)
- [23] F.L. Lewis, D. Vrabie, and V.L. Syrmos, *Optimal Control*, John Wiley & Sons, inc., 3rd edition, Hoboken, NJ, 2011.
- [24] T. L. Liu, “Optimal Aggressive Constrained Trajectory Synthesis and Control for Multi-Copters”, Ph.D. thesis, *The University of Texas at Arlington*, July, 2023.

## Biographical Statement

Suguru Sato was born in Takasaki, Gunma, Japan, in 1998. He visited NASA Kennedy Space Center when he was in high school, and this is when he decided to come to the U.S. to study Aerospace Engineering. After he graduated from a high school in Japan, he came to the United States to follow his passion, and received his bachelors degree in Aerospace Engineering from The University of Texas at Arlington in August 2022. He continued his academics at UTA to pursuit the masters degree in the same field. He worked as a Graduate Teaching Assistant for Flight Dynamics and Introduction to Aerospace Engineering courses. His current research interests include control of swarm of autonomous vehicles such as UAVs. After receiving his masters degree in Aerospace Engineering, he will pursuit his Ph.D. degree at UTA under current supervising professor Dr. Kamesh Subbarao.