A Novel Coordination Framework for Multi-Robot Systems

by

MOHAMMED TALEB ZAMZAM

Submitted to
the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

SABANCI UNIVERSITY

July, 2019

A Novel Coordination Framework for Multi-Robot Systems
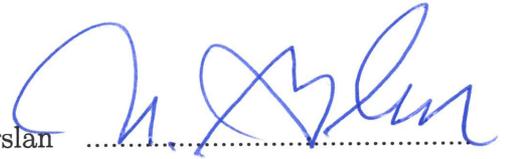
Mohammed Taleb Zamzam

APPROVED BY

Prof. Dr. Mustafa Ünel   .....Mustafa.....Ünel.....
(Thesis Advisor)

Assoc. Prof. Dr. Kemalettin Erbatur   .....Kemalettin Erbatur.....

Assist. Prof. Dr. Mehmet Selçuk Arslan   ....

DATE OF APPROVAL: ....16/7/2019........

A Novel Coordination Framework for Multi-Robot Systems

Mohammed Taleb Zamzam

ME, Master's Thesis, 2019

Thesis Advisor: Prof. Dr. Mustafa Ünel

Keywords: Multi-Robot Systems, Coordination, Trajectory Planning, Formation Control, AMRs, UGV, UAV, Quadrotor, Virtual Shells

Abstract

Having made great progress tackling the basic problems concerning single-robot systems, many researchers shifted their focus towards the study of multi-robot systems (MRS). MRS were shortly found to be a perfect fit for tasks considered to be hard, complex or even impossible for a single robot to perform, e.g. spatially separate tasks. One core research problem of MRS is robots' coordinated motion planning and control. Artificial potential fields (APFs) and virtual spring-damper bonds are among the most commonly used models to attack the trajectory planning problem of MRS coordination. However, although mathematically sound, these approaches fail to guarantee inter-robot collision-free path generation. This is particularly the case when robots' dynamics, nonholonomic constraints and complex geometry are taken into account.

In this thesis, a novel bio-inspired collision avoidance framework via virtual shells is proposed and augmented into the high-level trajectory planner. Safe trajectories can hence be generated for the low-level controllers to track. Motion control is handled by the design of hierarchical controllers which utilize virtual inputs. Several distinct coordinated task scenarios for 2D and 3D environments are presented as a proof of concept. Simulations are conducted with groups of three, four, five and ten nonholonomic mobile robots as well as groups of three and five quadrotor UAVs. The performance of the overall improved coordination structure is verified with very promising results.

Çoklu Robot Sistemleri için Orijinal bir Koordinasyon Yapısı

Mohammed Taleb Zamzam

ME, Master Tezi, 2019

Tez Danışmanı: Prof. Dr. Mustafa Ünel

Anahtar kelimeler: Çoklu Robot Sistemleri, Koordinasyon, Yörünge Planlaması, Oluşum Kontrolü, Otonom Mobil Robotlar İnsansız Kara Araçları , İHA, Dört-Rotor Sanal Kabuklar

Özet

Tek robotlu sistemlerin temel problemlerini çözme konusunda kaydedilen büyük ilerlemeler ile birlikte, birçok araştırmacı odağını, çoklu robot sistemleri (ÇRS) çalışmalarına kaydırdı. Hemen ardından, ÇRS'lerin, tek bir robotun yapması zor, karmaşık veya imkansız olduğu sayılan bazı görevler için mükemmel bir seçim olduğu tespit edildi örnek: mekansal olarak ayrı görevler. Bir grup robotun koordineli hareketinin planılaması ve kontrolü, ÇRS'lerin temel bir araştırma problemidir. ÇRS'lerin yörünge planlama probleminin çözümüne yönelik en sık kullanılan modeller arasında yapay potansiyel alanları ve sanal yay-damper bağları yer almaktadır. Bu metotlar matematiksel olarak sağlam sayılsa da, ürettikleri yollar, robotlar arası çarpışmaların engellendiğini garanti etmemektedir. Bu durum özellikle de robotların dinamikleri, holonomik olmayan kısıtlamaları ve karmaşık geometrileri dikkate alındığında geçerlidir.

Bu tezde, sanal kabukları kullanarak robotlar arası çarpışmayı önleyen yeni ve biyo-ilhamlı bir sistem sunulmaktadır. Bu sistem yüksek seviyeli yörünge planlayıcıya entegre edilmiştir. Bu sayede, düşük seviyeli kontrollerin izleyebileceği güvenli yörüngeler üretilebilmektedir. Robotların hareket kontrolü, tasarlanan hiyerarşik sanal girişler tabanlı kontroller ile sağlanmıştır. Kavram kanıtlama sürecinde kullanılan, iki ve üç boyutlu ortamlarda çeşitli koordineli görev senaryoları sunulmuştur. Üç, dört, beş ve on holonomik olmayan mobil robot grupları ile üç ve beş quadrotor İHA grupları üzerinde simülasyonlar gerçekleştirilmiştir. Geliştirilen genel koordinasyon yapısının performansı, ümit verici sonuçları ile doğrulanmıştır.

I dedicate this work to my most precious of all, my father Ali Zamzam.

To the memory of Mustafa Alsheikh Ali and Abdulabaset Sarut, On behalf of all dignity seekers, freedom fighters, martyrs, detainees, the oppressed... to all those who sacrifized...

Acknowledgements

I would like to take this oppurtunity to express my overwhelming emotions of admiration, appreciation and gratitude to my thesis advisor Prof. Dr. Mustafa Ünel for his exceptional supervision, enlightening guidance, and endless support. Since our first meeting together, I found in him the rare combination of an exceptional academician and caring elder brother in my expatriate. His passionate lecturing style, dedication to academia and seek for authenticity and novality will always be a role model for me as a researcher and an academic enthusiast.

I would also like to thank Assoc. Prof. Dr. Kemalettin Erbatur and Assist. Prof. Dr. Mehmet Selçuk Arslan for kindly spending their valuable time as my jurors.

I am thankful to my colleagues of the Control, Vision and Robotics (CVR) group, Gökhan Alcan, Diyar Khalis Bilal, Hammad Zaki, Naida Fetic, Mehmet Emin Mumcuoğlu and Emre Yilmaz for their support throughout my MS study. I specially thank Naida Fetic for her kind support during thesis writing process.

My graduate life would have been monotonously tiring if it wasn't for the good friends. I would like to thank my friends at FENS 1114 lab, Doğukan Kaygusuz, my great room-mate at the same time, Hande Karamahmutoğlu, Sümeyra Vural, Enver Ersen, Ege Can Onal and H. Yusuf Altun for providing such a warm and friendly environment. I would also want to thank my bro Zain Fuad and my brother, ex room-mate İsmail Al Haddan who graduated last year. I extend my thanks to all my friends in my off campus life.

I am also extremely grateful to my beloved family for being everything in my life. My father with his selfless hard work for our sake, my caring mother that makes me feel calm and happy when she is around and my precious brothers Abd Alrahman and Omar.

My last thought goes towards my one and only love, my lovely wife Fatima Alsheikh Ali. I would like to thank her for carrying me through the tough times, removing all the stress I brought home with positive energy and delicious food. I feel blessed with her being in my life with all the unconditional patience and support that I knew I could always count on, all while never asking for anything in return.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

A single-robot system contains only one individual robot that is able to model itself, perceive its environment and model their mutual interactions [1]. Soon after reaching a satisfactory level of individual stationary robots usage in industry, the interest towards mobile robots began to increase. Furthermore, advances in electronics and consequently the increased computational capacity pushed towards the notion of autonomy in mobile robots. Along with other factors, this led to the emergence of autonomous mobile robots (AMRs) research field. AMRs can be categorized into three main groups: unmanned ground vehicles (UGVs), unmanned marine vehicles (UMVs), also known as autonomous underwater vehicles (AUVs), and unmanned aerial vehicles (UAVs). Since Shakey [2], the first general-purpose mobile robot platform, several individual robot projects had been developed and become well-known such as ASIMO [3], BigDog [4] and PR-2 [5].

<div align="center">(a)   (b)   (c)</div>
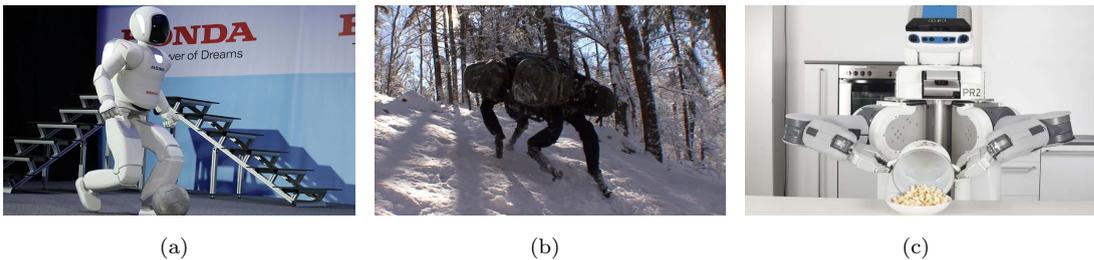
FIGURE 1.1: Well-known single-robot projects: (a) ASIMO, (b) BigDog and (c) PR-2

AMRs are suited for tasks that are considered to be dull, dirty and dangerous for humans. Despite that single-robot systems have shown a relatively strong performance, some tasks are too hard, complex or even impossible for it to perform. On one hand, this might be due to robot-dependent factors such as its power limitations, locomotion mechanism and design constraints. On the other hand, it relates to the task nature itself, e.g. spatially separate tasks such as surveillance, air-ground combat, area coverage and exploration missions.

Instead of designing a highly sophisticated and expensive single robot, research community was attracted to the idea of using cheaper and simpler robot groups that can accomplish exactly the same tasks cooperatively [6]. Accordingly, researchers from various disciplines gradually shifted their focus from the ordinary single-robot systems towards multi-robot systems (MRS) along with their versatile applications and interesting challenges.

Research efforts investigating MRS began as early as the 1990s and have been rapidly growing ever since. A series of group robot projects have been lunched such as GOFER [7], CEBOT [8], M+ [9] and ASyMTRe [10]. MRS can be effective in tasks such as exploration, search and rescue, unknown and partially known environments' mapping, reconnaissance remote sensing, hazard identification and removal. In practice, MRS already made its way to industrial and commercial use, e.g. Warehouse UGV robots and goods delivery network by drone UAVs as can be shown in Fig. 1.2.

(a)

(b)

FIGURE 1.2: Industrial and commercial-use MRS: (a) Amazon's warehouse robots, (b) Cargo delivery by drone

MRS can have several potential advantages over single-robot systems:

- Better overall system performance, with task execution time and total energy consumption as evaluation metrics [11, 12].

- Wider spatial distribution.

- Enhanced flexibility, reliability, scalability [13] and versatility.

- Cost efficiency; when a single complex expensive robot that is compared to a group of simple cheap group of robots can do exactly the same task.

The term *coordinated motion* denotes the motion of a MRS's robots to accomplish a predefined task in coordination. The motion of each robot member is dependent on the motion of others in the group. In addition, coordination inherently assumes moving in some desired formation. The necessary formation varies according to the coordinated task [14]. Despite remarkable research developments in the area, numerous challenges remain. These challenges include designing appropriate coordination strategies, inter-robot communication, relative state sensing and estimation, control paradigms appropriate to real-time systems, fusion of distributed sensors data, task allocation, path planning, formation maintenance and obstacle avoidance.

One essential problem of interest is the inter-robot collisions. Motion in formation levels up the risk of collisions especially while constructing the formation or while interacting with other elements in the environment. The autonomous robots forming the group must avoid collisions with each other without sabotaging the overall mission. Despite the existence of several mathematical models that are *sound and complete* in the sense that the robots converge to the predetermined goal, this turns out to be one of the non-trivial problems [15]. The robots should change their path to avoid collisions even if this will introduce some delay in the coordinated task achievement.

1.1 Problem Formulation

We hereby explain the coordinated task that will be used hereinafter as a performance criteria for our different test bed robot implementations.

The task encompasses a group of n autonomous mobile robots, namely, $R_1, R_2, \ldots R_i, \ldots, R_n$ and an object, G, that will be considered as a goal modeled as a point at its center of mass. The coordinated task scenario can be divided into three sub-tasks summarized as follows:

1. Robots $R_1, R_2, \ldots R_i, \ldots, R_n$ converge to a circular formation of a predefined radius d_G positioned around the goal G point regardless of the initial configuration of both robots and goal.

2. The robots will move in a coordinated fashion i.e. maintaining a predefined mutual distances d_R, hence forming a triangular mesh, while approaching G.

3. Generated trajectories for the robots need to be safe. Contact/overlap between two or more robots should not be allowed during the execution of the coordinated task, i.e. inter-robot collisions must be avoided.

Initial configuration is assumed to be a priori given including initial positions of the robots and the stationary goal's position. Robots are also assumed to communicate their positions and velocities by some communication protocol or perception capabilities. The design of such protocol is not trivial as will be discussed later in detail, but it is out of the scope of this work.

Fig. 1.3 illustrates a 2D homogeneous robots based example of the task in its three phases. One possible realistic application is the task of surrounding an enemy target by a group of homogeneous/heterogeneous robots to prevent it from escaping and force it to surrender. Moreover, higher complexity tasks can be partitioned into a group of simple sequential tasks some of which is the the one proposed here.
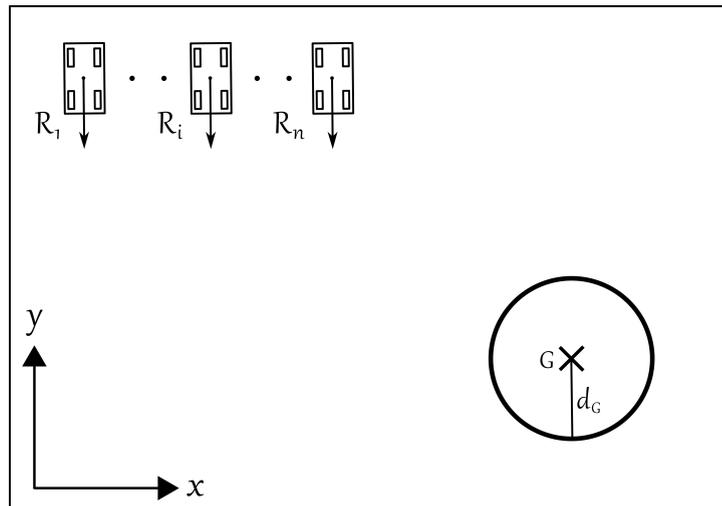
(a)
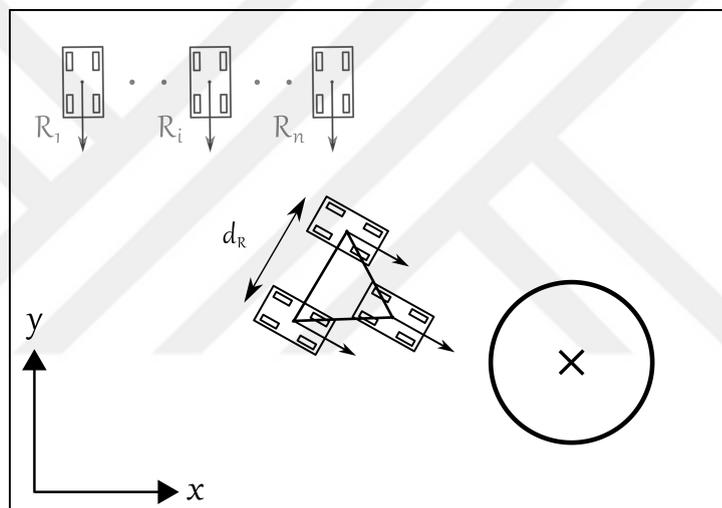
(b)
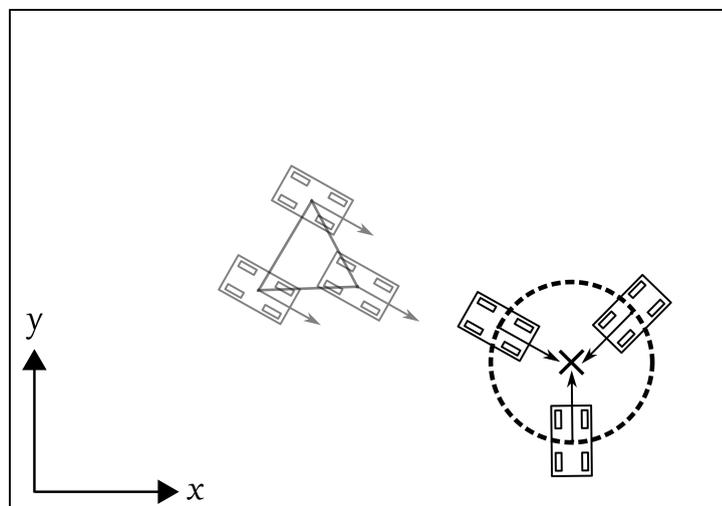
(c)

FIGURE 1.3: Coordinated task scenario (a) Initial configuration, (b) Approaching goal in a coordinated manner and (c) accomplishing the task

1.2   Thesis Contributions

The contributions of this thesis can be summarized as follows:

- A novel framework inspired by animal swarms in nature to solve the collision avoidance problem of MRS is proposed. Virtual shells concept is established and utilized as its mathematical model. The newly elaborated coordination structure is proved to guarantee the online generation of collision-free trajectories for every individual robot member of the group by means of various computer simulations.

- An algorithm motivated by rigid body elastic collisions is developed as a solution to the collision response sub-problem. Additionally, an alternative abstract algorithm that necessitate less communication requirements on the expense of efficiency is also developed. Both algorithms are analyzed in detail and compared for different collision scenarios.

- Position and attitude trajectory controllers for nonholonomic UGV robots' low-level motion control are designed using virtual inputs hierarchical control approach.

- Results are successfully verified in simulation environment for two distinct MRS setups; (2D) including groups of three, four, five and ten nonholonomic mobile robots, and (3D) with groups of three and five quadrotor UAVs.

1.3 Thesis Outline and Organization

The remainder of the thesis is organized as follows:

Chapter 2 provides a literature survey on multi-robot coordination and formation control including the commonly utilized configurations. It also gives a background on some well-known mathematical models used to solve coordination problem as well as a brief review on the collision avoidance problem.

A planner scheme for reference trajectory generation of multi-robot systems is adopted from previous works in the literature, briefly explained and customized for this thesis contribution means inn **Chapter 3**. The used model is first established for 2D, and then extended to 3D MRS with the help of polynomial trajectories.

Chapter 4 introduces the kinematic model of nonholonomic mobile robots and the kinematics and nonlinear dynamic model for the quadrotor UAVs. Hierarchical control is developed for the position and attitude of each of the two models using virtual inputs control concept.

In **Chapter 5**, The concept of virtual shells upon which the contributed collision avoidance framework is built, is unfolded. Shell's geometry, collision detection and collision response sub-problems are formulated and discussed in detail. Elastic collision (EC) model and light beam reflection (LBR) model are proposed, formulated and analysed.

Chapter 6 presents the simulation results of the proposed framework. Rich set of scenarios are established and carried out in simulation environment on groups of three, four, five and ten nonholonomic robots as well as groups of three and five quadrotor UAV. Different collision scenarios are particularly emphasized, and controller's performance is verified by means of desired vs. actual state graphs and Root Mean Square Error (RMSE) evaluation. Scenario-based comments are made to discuss the achieved results.

Thesis is concluded with several remarks in **Chapter 7** and possible future research directions are indicated.

Chapter 2

Literature Survey and Background on Multi-Robot Coordination

In a multi-robot systems, the development of models systematically describing the motion of each robot member as well as the group as a whole is non-trivial. Researchers from different disciplines have been putting efforts recently towards tackling this problem in particular [16–20].

This chapter outlines various architectures developed in the literature to attack this problem. These theoretical classifications are coupled with mathematical tools and models that fulfill its conditions and objectives of coordinated motion. We review some of the most widespread models, namely, artificial potential fields (APFs) and virtual spring-damper bonds as a basis for the coordination framework.

Finally, collision avoidance sub-problem is briefly reviewed. Various attempts to tackle this problem in the literature are particularly emphasized to elaborate on and contribute into in chapter 5.

2.1 Coordination Configurations

We present the major coordination paradigms that should be considered while formulating the problem. This includes centralized, decentralized decision making configurations and virtual structure abstraction. We briefly review efforts in the literature utilizing such configurations in multi-robot coordination tasks.

Other configurations are also available but are not reviewed as they are out of scope for this thesis, e.g., graph theory based coordination [21], [22], [23] and non cooperative systems' coordination via robotic herders [24, 25] .

2.1.1 Centralized Systems

Centralized systems are a natural extension from single robot systems. In centralized decision-making control scheme, one or more of the robots are considered to have leadership status i.e. designated as leader(s), having global information about the environment. Leaders can communicate main navigation information to other robots but they cannot receive information from them. On the other hand, other so-called follower robots can transmit and receive data. Such configuration is called a leader(s)-follower(s) formation control [26, 27].

A direct shortcoming of such architecture is its lack of robustness against dynamic environments and failures. Its ultimate dependence on the leader makes it prone to system failures whenever the leader is defected [19]. Nevertheless, among many advantages of such scheme, one should bring up its ability to be modeled as a whole in an exact fashion, thus making globally optimal plans producible [28].

Different versions of this scheme have been presented in the literature [29]. This includes Leader-Obstacle Configuration, where the follower robot performs some self-behaviors such as avoiding obstacles in its sensing region while still following the leader, thus having a decentralized theme on follower level.

Original works in multi-robot systems utilizes leader-follower configuration. For instance, nonholonomic mobile robots motion coordination is investigated in [30]

using leader-follower configuration combined with virtual robot and reference trajectory generation concept. In this work, leader provided position and heading information while its velocity state was estimated using a designated observer. Followers track the reference trajectories using integral back-stepping controller. As regarding collision avoidance and obstacle avoidance sub-problems, authors integrate a fuzzy logic based algorithm by sensing the relative distance between follower robots and obstacles.

In [31], a leader-follower formation control approach for AUVs is proposed. Due to underwater communication constraints, only leaders' positions are sent to followers and reference trajectories are generated with respect to a predefined distance. Back-stepping control and Lyapunov analysis is performed to ensure trajectory tracking.

Dierks et al. proposed a quadrotor UAVs composed MRS coordination method based on spherical coordinates in [32]. The aim is for the follower UAV to track its leader at a desired separation, incidence angle and a bearing. Moreover, neural network (NN) based control law that learns the dynamics of UAV is presented. Stability of the formation for unmodeled disturbance such as aerodynamic friction is proved via Lyapunov theory.

Trajectory tracking and flight formation for quadrotor UAVs is tackled in [33] and a solution based on leader-follower scheme is proposed. Time scale based control separation is made where translational dynamics are controlled using a sliding mode controller, while desired orientation is maintained using a linear PD controller. A third controller is designed for the follower to solve the formation problem in horizontal plane.

2.1.2 Decentralized Systems

Decentralized control approaches can be further divided into two categories: hierarchical and distributed. Distributed decentralized architecture is inspired by the behaviors of animals, where every stimuli has its corresponding response (behavior) [34]. Flocking birds, schooling fish and bees are examples of group members working in high coordination with others in their group achieving a task without depending on online orders from some high-level "leader" [16]. In robotics, the same approach is used to control MRS autonomously, in coordination and without the need for a global control over the system.

FIGURE 2.1: Decentralized animal swarms: (a) flocking birds, (b) ant swarm

An objective in a decentralized multi-robot system is decomposed into independent sub-problems. For the system to achieve that goal objective, each individual in the system needs to follow some local rules in order to accomplish every sub-problem accordingly. These rule bundles apply simultaneously on robot-level and have parallel access on its perception mechanisms. One shortcoming is that, on the contrary to centralized systems, agents cannot predict group's overall behavior with only the local information in hand. Consequently, some group behaviors cannot be controlled.

Research efforts in this field revealed that, there are many coordinated motion tasks that can be done more efficiently and robustly using decentralized multi-robot systems [14]. Its scalability, parallelism, robustness and computational efficiency together with other properties motivated further work.

In [35] for example, authors implement a decentralized control on three autonomous robots to navigate with a triangle formation and maintain it while avoiding obstacles at the same time. They consider non-linear dynamical systems and use attracting and repelling vector fields as the mathematical tool to achieve system's objective. Li et al. [36], propose an algorithm for AUVs MRS that uses potential field layers for tasks such as formation control and obstacle avoidance in an uncertain environment.

Ghose et al. proposed altitude and heading angle consensus for leaderless but connected nonlinear UAV swarm [37]. It is based on several sliding mode controllers. Authors also present sliding mode control-based autopilots that allow for individual members to fly independently. They prove asymptotic stability for the controllers and autopilots as well as system's insensitivity to disturbances and parameter variations while controlled.

Through out the rest of this thesis, we adopt distributed decentralized architecture in our multi-robot system. This is due to its appealing attributes that includes but are not restricted to:

- An individual robot failure in a decentralized system does not necessarily terminate the overall mission, unlike the case in centralized systems.

- Decentralized systems are proved to outclass centralized systems in particular tasks of interest such as area coverage, exploration, surveillance and search and rescue activities [38].

- Decentralized systems usually are of a low cost when compared to centralized versions carrying out the same objective [39].

A thorough comparison between centralized leader-follower configuration and decentralized schemes in [19, 39–41].

2.1.3 Virtual Structure Configuration

Virtual structure configuration implies treating the formation as a rigid body in the sense that it has its own reference frame. Positions of robots in this virtually synthesized structure are defined as position vectors measured from reference frame's origin e.g. geometrical center of the formation. Assuming reference point has its own offline or online planned trajectory, desired trajectory for each robot in the structure can be easily found by simple vector combinations. Motion controller is then responsible of guarantee desired trajectories tracking [42].

In [43], Nijimeijer et al. tackle the problem of formation control for unicycle mobile robots. Virtual structure controller that uses mutual coupling between individual robots is designed as it makes it more robust against perturbations in comparison with leader-follower configuration. Similarly, in [44], authors propose a virtual structure formation for UAVs in 3D space with corresponding tracking approaches. UAVs can track desired formations even when the structure moves slower than their minimum speed. To minimize the risk of crashing while constructing a formation they utilize deconfliction controller.

A recent work by Schwager et al. [45] combines different concepts to achieve an agile coordination and collision avoidance for a swarm of quadrotor UAVs. Authors use Virtual Rigid Body (VRB) abstraction to plan trajectories for formation maintenance and transitioning between different formations. Virtual structure concept was integrated with differential flatness based feedback control for every quadrotor to track its trajectory in the formation. This allowed the swarm to be teleoperated as if it was a single quadrotor eventually causing the framework to be scalable for an arbitrary number of quadrotors. Multiple layered potential fields were implemented to perform tasks such as static obstacle collision avoidance, quadrotor collision avoidance and formation hold. They performed a successful 200 quadrotors swarm simulation and physically implemented their algorithms with formations of 5 quadrotor UAVs.

2.2 Mathematical Modeling Approaches

Coordination paradigms are implemented with the assistance of different mathematical models. They enforce robot-level local rules in such a way that the necessary behavior(s) are implemented. Two of the widely-used mathematical tools, among many others, are artificial potential fields and virtual viscoelastic forces. We briefly go over the basis of these tools as a preliminary other work in this thesis. We also cite example works and express existing limitations.

2.2.1 Artificial Potential Fields (APFs)

This concept was first introduced in 1986 by Khatib et al. [46] as a real-time obstacle avoidance algorithm. It is based on synthesizing circular virtual potential fields around robots, obstacles and goal all considered as points. Robots in the system are supposed to navigate following the global velocity vector field. Robot(s) navigation with the help of potential fields has been extensively used by researchers [45, 47–49].

As Vlantis et al. did in their work [49], consider a test robot R_i of state vector X_i modeled as a single integrator in a n-dimensional workspace. Its model can then be mathematically described as : $\qquad \dot{X}_i = u$ $\qquad\qquad\qquad$ (2.1)

where $\dot{X}_i$ is robot's velocity vector and u is the input. For a robot to navigate, it only needs to equate its velocity input to the negative gradient (steepest descent) of the resultant artificial potential field function:

$$u = -\nabla \Psi_{net} \qquad\qquad (2.2)$$

where Ψ_{net} is the net potential field function. We call it net/resultant as we use APFs in layers to perform different tasks (behaviors):

$$\Psi_{net} = \lambda_1 \Psi_1 + \lambda_2 \Psi_2 + \cdots + \lambda_k \Psi_k \qquad\qquad (2.3)$$

where each Ψ_k represents either an attracting or a repelling potential field layer aiming at some intended behavior and λ_k is the corresponding weight. Behaviors means different coordinated task objectives: target (goal) attraction or neighboring robots, obstacles and environment boundaries repulsion respectively.

One can define each potential function as:

$$\Psi_k == f_k(\cdot)(|X_i - X_j|) \tag{2.4}$$

where $f_k(\cdot)$ is a suitably defined function satisfying $f_k(r) \to 0$ as $r \to 0$, and X_j is the position of the other object; a target, an obstacle or another robot [47]. Figure 2.2 shows a single robot simulation results using potential fields.
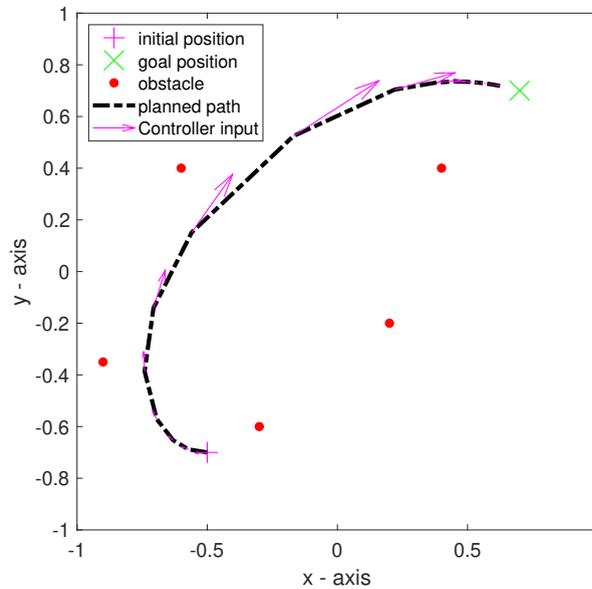
FIGURE 2.2: Artificial potential fields navigation implementation using a single robot

One major drawback of APFs is the presence of a local minima that might trap the robot. Many attempts to solve this problem has been made in the literature e.g. using world transformations, offline scenario dependent weight tuning and using minima free complex functions [50]. As a conclusion, despite its soundness, intuitive design and wide use, it is rarely used as it is. Typically, it is integrated with other technique.

2.2.2 Virtual Viscoelastic Bonds

Virtual springs-dampers is an intuitive idea of synthesizing virtual bonds between two or more objects for coordinated motion purposes. Springs-dampers have been used to model the connection between two or more masses; thus, adding elastic and viscous friction characteristics to that connection. This concept is used in many robotic fields: flexible robots [51], robotic manipulators [52], vibration modeling and suppression [53] and most importantly, in our context, multi-robot coordination path planning, formation maintenance and obstacle avoidance [54–56].

It uses external forces to shape a desired path for each robot in the swarm. Considering a simple double integrator dynamic model for a test robot R_i of mass m_i

$$m_i \ddot{X}_i = F_{net} \tag{2.5}$$

where $\ddot{X}_i$ is the acceleration of the robot R_i. Newton's second law, implies that $\sum F_{ext} = ma$ thus the motion of this robot is governed by the net force acting on it. As a planner, one can synthesize virtual forces that act on the robot and make their combination to be equal to F_{net}:

$$F_{net} = F_1 + F_2 + \cdots + F_k \tag{2.6}$$

As a result, robot R_i will be forced follow a trajectory, that will be my virtual reference trajectory to be tracked by the controller.

We think of these F_k forces to be either a *spring force* that works on maintaining an equilibrium distance between the robot and goal, obstacle or another robot's position, or a *viscous damping force* in which robot's velocity seek to converge to a reference velocity being either zero in the case of goal approaching or another robot's velocity when a coordinated motion of the group is being imposed [54].

2.3   Collision Avoidance

While robots are in motion trying to achieve a predefined task, it is crucial to handle cases where they collide. Two steps are generally necessary to avoid collisions:

- Collision Detection
- Collision Response

Collision response is also called "collision resolution" in the context of fields other than robotics e.g. computer graphics [57].

Even robust algorithms for multi-robot systems encounter the challenge of collisions. Both potential fields and virtual bonds are presumably capable of generating collision-free paths but in practice it is not the case. The problem even gets more complicated when nonholonomic constraints, nonlinear dynamics and complex geometries comes into the picture [54]. This is why many researchers augment these mathematical models with other algorithms thus eliminating collisions.

For instance, nonholonomic mobile robots' inter-robot collisions and obstacle avoidance sub-problems in [30] were solved by integrating a fuzzy logic based algorithm via sensing the relative distance between follower robots and obstacles. In an another work [36], authors indicate some disadvantages of obstacle avoidance using potential fields such as the creation of bigger avoidance radii and propose a region separation based technique to solve the problem.

Moreover, in order for the virtual spring-damper connections to guarantee inter-robot collisions-free paths for scenarios as in [56], a parameter tuning process of spring/damper coefficients is needed. This can be done by increasing the rigidity and decreasing flexibility of the bonds eventually negatively affecting coordinated motion quality, convergence speed to formation and possibly task completion. Gulec et al. [54], emphasized this problem and augmented his virtual spring-damper based coordination model with an additional online collision avoidance algorithm utilizing the idea of Virtual Collision Prediction Region (VCPR).

Chapter 3

High-Level Planner: Reference Trajectory Generation

The coordination framework have two levels interacting in a hierarchical way as depicted in Fig. 3.1. First, virtual reference trajectories are generated for every robot R_i in the group. Trajectories are designed such that they fulfill the formerly defined coordinated motion task. Physical robots then have to track their desired paths via designated controllers synthesized based on robot's kinematic and dynamic models. This chapter explains the planner level, whereas controller synthesis is explained in Chapter 4. We first establish the structure for 2D environments clarifying its component models, we then extend it to 3D setups.
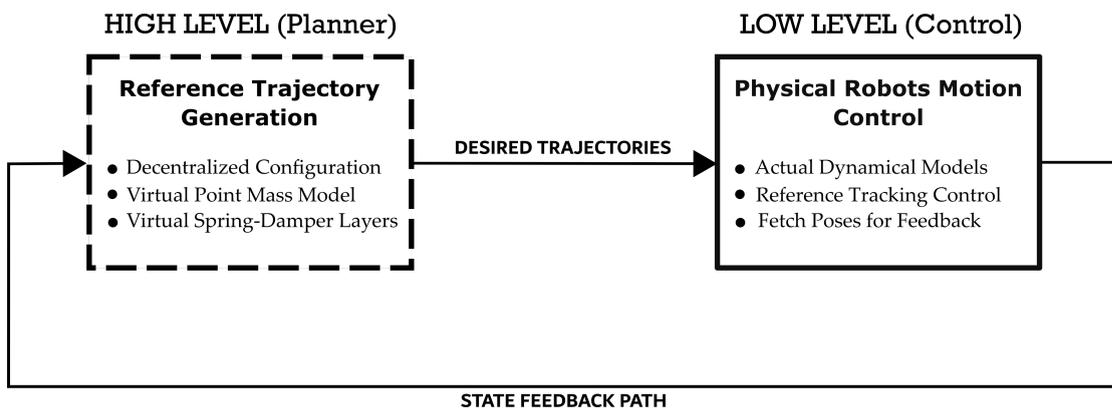
FIGURE 3.1: Hierarchical coordination structure

3.1 Virtual Point Masses

Since we are in the prior level of the planner, one smart abstraction comes from noting that we need not to consider complex, nonholonomic or nonlinear models. Thus, we choose to model reference trajectory generation system's robots as virtual point masses $m_1, m_2, \ldots m_i, \ldots, m_n$. For the model to mimic real-life scenarios we add a viscous friction term to the model limiting velocity profiles of the generated trajectories. For every virtual mass m_i the dynamic model is:

$$m_i \ddot{X}_i + b_i \dot{X}_i = F_{net} \tag{3.1}$$

where b_i is m_i's friction coefficient and $\ddot{X}_i$, $\dot{X}_i$ are its acceleration and velocity vectors respectively. The motion of each point mass is completely driven by the net force F_{net} acting on it. F_{net} can be treated as the linear combination of multiple distinct forces:

$$F_{net} = F_1 + F_2 + \cdots + F_k \tag{3.2}$$

Note that point masses model is holonomic, this approach relaxes the nonholonomic constraint. Thus, forces in action can move m_i in any direction. On the other hand, orientation is not defined. However, physical robots' controllers generally need a desired orientation to track. Thus, reference orientation will be obtained from the velocity profile of the virtual mass.

3.2 Virtual Spring-Damper Forces

We follow a decentralized distributed coordination configuration while establishing the system. The interest is to implant a bundle of local rules in every individual virtual mass in such a way that both: its own motion, and the group as whole will exhibit the intended *behaviors*. These rules will be mathematically interpreted as virtual forces $F_1, F_2, \ldots, F_k$ composing F_{net}. These forces should depend on the virtual mass's parameters (position, velocity, mass) as well as other masses.

In this section, we design the necessary and sufficient virtual forces F_k for the generated trajectory to fulfill the coordinated motion problem. We synthesize two

layers of virtual spring-damper bonds as presented in previous works [54, 56, 58]. Viscoelastic forces connecting rigid bodies come in pairs. Spring force, virtual in our case, is responsible for maintaining a predefined distance that the connected bodies converge to elastically. On the other hand, virtual damper force will rigidly connect them so that they move together, i.e. level their velocity vectors to a common one. Together, they establish a foundation for the coordinated motion and formation control of rigid bodies.

3.2.1 Layer 1: Mass-Mass Virtual Forces

The coordinated motion of the virtual masses and their formation control are achieved in this layer with pairs of virtual spring-damper connections. While developing the necessary forces for these sub-tasks we assume that every m_i perceives and tries to coordinate only with its two *Nearest Neighbors*, a concept first introduced by Vicsek et al. in 1995. For instance, m_i does not form any virtual bond with m_{j_3} as in Fig. 3.5.

A virtual force F_1 acting on m_i is then designed as follows:

$$F_1 = - \left[k_R(d_{i \to j_1} - d_R) + c_R((\dot{X}_i - \dot{X}_{j_1})) \cdot u_{i \to j_1} \right] u_{i \to j_1}$$
$$- \left[k_R(d_{i \to j_2} - d_R) + c_R((\dot{X}_i - \dot{X}_{j_2})) \cdot u_{i \to j_2} \right] u_{i \to j_2} \qquad (3.3)$$

where $\cdot$ denotes a dot product of vectors, k_R and c_R are spring-damper coefficients, respectively, $u_{i \to j1}$ and $u_{i \to j2}$ are unit vector from m_i to m_{j_1}, m_{j_2} while $d_{i \to j_1}$ and $d_{i \to j_2}$ are their corresponding distances. $\dot{X}_i$, $\dot{X}_{j_1}$ and $\dot{X}_{j_2}$ are the velocity vectors of m_i and m_{j_1}, m_{j_2} respectively, d_R is the distance to be maintained between m_i and m_{j_1}, m_{j_2}.

F_1 is a combination of a virtual spring-damper forces due to m_{j_1} and another pair due to m_{j_2}. Therefore, m_i always try to maintain a distance d_R between itself and each neighbor mass but at the same time its motion is affected by its neighbors velocity vectors magnitude and direction, i.e. it moves with them.
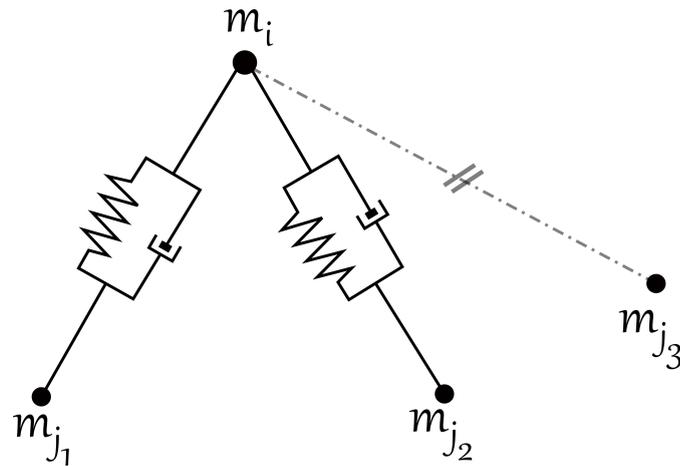
FIGURE 3.2: Virtual bonds between m_i and its two nearest neighbors m_{j_1}, m_{j_2}

When three masses close to each other are connected with virtual bonds, they mutually act on each other with F_1. We, eventually, achieve the uniform triangle-shaped formation with their motion also connected as depicted in Fig. 3.3.
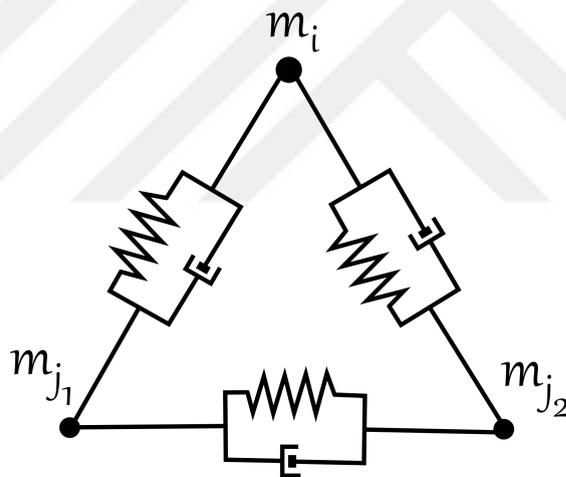
FIGURE 3.3: Layer1: three virtual masses mutually connected by virtual bonds

When every mass is acted upon by this force considering its two neighbor robots, the group will eventually converge to a triangular mesh formation composed of fundamental groups of three masses in a triangle shape. The developed structure is thus modular and can be scaled to n number of virtual masses consequently, to n robot. For the sake of clarity, we restrict our system description to its most elemental module, three virtual masses. We show the scalability property performance with simulations in Chapter 6.

3.2.2 Layer 2: Mass-Goal Virtual Forces

This layer is intended to force each mass m_i to converge to a predefined distance around the goal G. A virtual spring-damper pair is attached between the virtual mass m_i and the goal G as in Fig. 3.4.
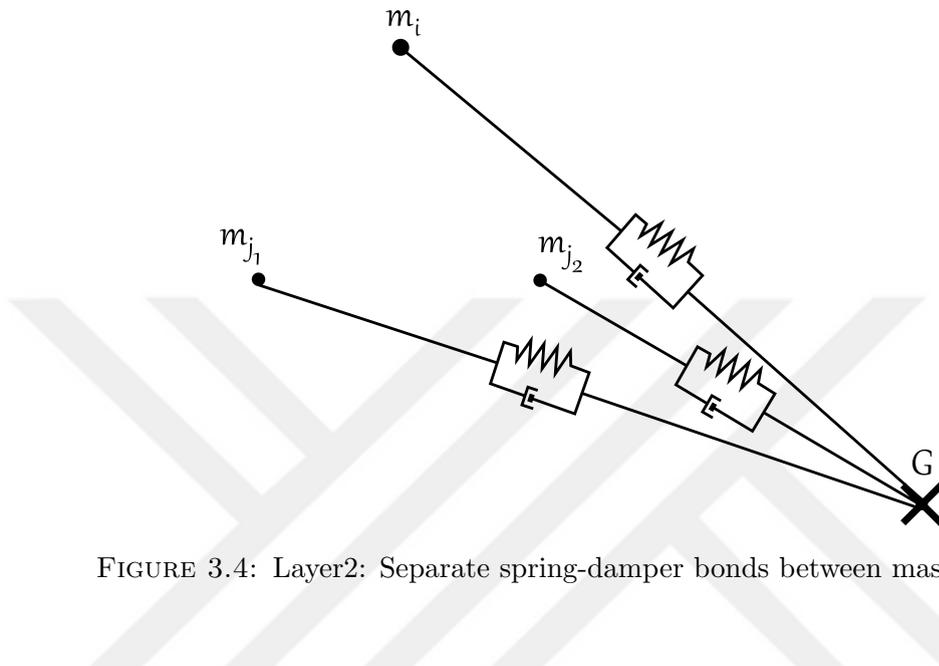
FIGURE 3.4: Layer2: Separate spring-damper bonds between masses and G

For this purpose, A force F_2 is synthesized as the combination of pair of spring-damper virtual forces as follows:

$$F_2 = -\left[k_G(d_{i \to G} - d_G) + c_G(\dot{X}_i \cdot u_{i \to G}) \right] u_{i \to G} \tag{3.4}$$

where $\cdot$ denotes a dot product of vectors, k_G and c_G are spring-damper coefficients, respectively, $u_{i \to G}$ is the unit vector from m_i to G while $d_{i \to G}$ is the distance between them. $\dot{X}_i = [\dot{x}_i, \dot{y}_i]^t$ is the velocity vector of m_i and d_G is the distance to be maintained around G.

F_2 acts independently on each m_i, its spring force component attracts it towards a distance d_G from G. Damper force component, however, have an equilibrium velocity of *zero*; thus, it decreases m_i's velocity while approaching until it is eventually rested on the desired spot defined by spring force.

Uniform Formation around G

The development above guarantees the convergence of every mass m_i to a distance d_G from G. However, in order for the formation to lie perfectly on a circle of radius d_G centered at G, the formation distance between approaching robots d_R have to be a function of d_G derived from simple trigonometry. This is an unnecessary constraint as it will limit our choice to one possible d_R. We bypass it using the approach proposed in [54] in which an adaptable model parameter laws are triggered whenever a mass m_i enters a concurrent circular region defined by a radius d_{break}. We are therefore free to select desired d_R during goal approaching phase, While a strict value denoted by d_R' depending on d_G and n, the number of virtual masses, is applied as the robots reach d_{break} region.

Robots' inter-distance necessary for a uniform circular formation is changed after passing d_{break} to d_R' according to the following trigonometric relation:

$$d_R' = d_G\sqrt{2(1 - cos(2\pi/n))} \tag{3.5}$$

Spring coefficient, k_G, is also changed to smaller value of k_G' thus relaxing the
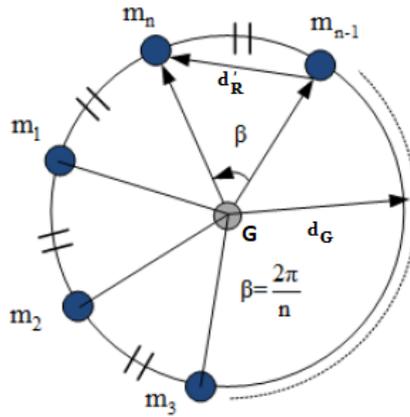
FIGURE 3.5: Uniform circular formation of masses around G

connections in an aid to the formation process. Authors in [54] develop a sigmoid function to guarantee a smooth continuous transition:

$$k_{Robots} = k_R' + \frac{k_R - k_R'}{1 + e^{\alpha(d_G' - d_{i \to G} + \gamma)}} \tag{3.6}$$

Figure 3.6 illustrates the final configuration where both layers are combined constructing the high-level planner.
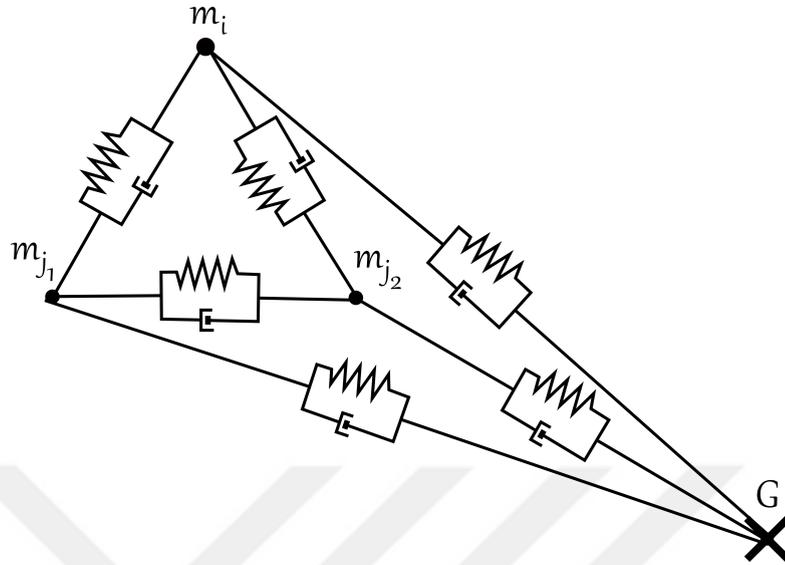
FIGURE 3.6: Full virtual spring-damper forces scheme

When F_{net} in acting on m_i is equal to $F_1 + F_2$, equation 3.1 becomes:

$$m_i \ddot{X}_i + b_i \dot{X}_i = F_1 + F_2 \tag{3.7}$$

The resultant set of $\left[X_i, \dot{X}_i, \ddot{X}_i \right]$ obtained from each m_i model are the desired reference trajectories for every corresponding robot R_i. As for the desired orientation profile, it can be easily derived from velocity vector components $\dot{X}_i$:

$$\theta_i = \arctan \left(\frac{\dot{y}_i}{\dot{x}_i} \right) \tag{3.8}$$

Collision Avoidance

Unfortunately, the previous formulation alone is not sufficient to generate collision-free paths and implicit solutions are impractical and scenario-dependent as discussed in details in Chap. 2. We establish our proposed scenario-independent collision avoidance framework, develop its mathematical representation and integrate it with this virtual spring-damper based scheme later in Chap. 5. As a result, the enhanced overall coordination framework successfully performs all sub-tasks of the formulated problem.

3.3   Extension to 3D Environments

The previous models were developed for 2D workspace. Despite that we can easily extend them to higher dimensions workspaces, it might not be the most fitting solution. For example, for a heterogeneous MRS of UGVs and UAVs working together, UGVs will have upward force component which is not feasible. Instead, we append a suitable desired trajectory for z dimension for each robot R_i that encompasses the set $\left[Z_i, \dot{Z}_i, \ddot{Z}_i \right]$ to have a full 3D trajectories. By this abstraction, the whole coordination and formation control process happens on xy horizontal plane, and the Z trajectory have the sole responsibility of smoothly lifting the UAVs to the desired elevation as depicted in Fig. 3.7. Obvious shortcoming of this methodology is not benefiting from the spatial versatility of the 3D space for different mid-air 3D formations.

For that purpose, we use the concept of polynomial trajectory generation [59].
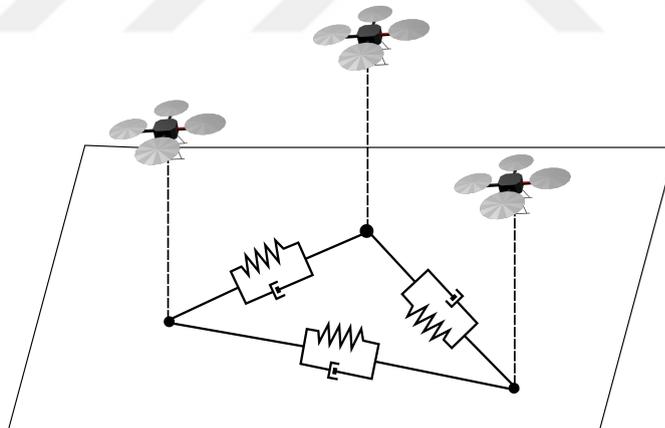
FIGURE 3.7: 3D spring-damper connections configuration

3.3.1   Polynomial Trajectory Generation

The quadrotor UAV's dynamic model is proved to be deferentially flat. Moreover, all four times continuously differentiable, known as C^4, paths are proved to be dynamically feasible with the proper controller design [60].

We design a quintic (5th order) polynomial trajectory for two main reasons: It is four times continuously differentiable, and we can select initial and final points positions, velocities and accelerations:

$$P = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \tag{3.9}$$

We define our desired initial and final z axis positions' parameters $[z_i, \dot{z}_i, \ddot{z}_i, z_f, \dot{z}_f, \ddot{z}_f]$ together with initial and final time values $[t_i, t_f]$. The polynomial trajectory's coefficients can then be found by:

$$
\begin{pmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix}
=
\begin{pmatrix}
t_i^5 & t_i^4 & t_i^3 & t_i^2 & t_i & 1 \\
5t_i^4 & 4t_i^3 & 3t_i^2 & 2t_i & 1 & 0 \\
20t_i^3 & 12t_i^2 & 6t_i & 2 & 0 & 0 \\
t_f^5 & t_f^4 & t_f^3 & t_f^2 & t_f & 1 \\
5t_f^4 & 4t_f^3 & 3t_f^2 & 2t_f & 1 & 0 \\
20t_f^3 & 12t_f^2 & 6t_f & 2 & 0 & 0
\end{pmatrix}^{-1}
\cdot
\begin{pmatrix} z_i \\ \dot{z}_i \\ \ddot{z}_i \\ z_f \\ \dot{z}_f \\ \ddot{z}_f \end{pmatrix}
\tag{3.10}
$$

Our Z trajectory profile is simply P, and we can get $\dot{Z}, \ddot{Z}$ from differentiating it accordingly.

Chapter 4

Low-Level Motion Control: Physical Robots Modeling and Control

The planner-generated trajectories need to be tracked by the physical robots in order to achieve the coordinated task. This is non-trivial since the planned trajectories are holonomic, however, almost non of the physical robots are. Moreover, the produced dynamically feasible paths was not particularly emphasized, thus a proper synthesis is needed for different kinematic and dynamic model. In this chapter, the modeling and control of two different robot types who will be used as a validation test bed in this thesis: nonholonomic mobile robots and quadrotor type UAV.

4.1   Nonholonomic UGV

Nonholonomic robots are among the most used robot models especially in MRS coordination context. This is because they are low-cost and abundantly available in the market. The nonholonomic constraint indicates restrictions on directions of motion. This restriction complicates the mathematical representation of the

system and consequently synthesis. The control for such type of robots is well-investigated in the literature [61–63]. In this subsection, unicycle type nonholonomic mobile robots model is reviewed and a hierarchical scheme is developed using virtual inputs in order for the robots to track their trajectories.

4.1.1 Kinematic Model for UGVs

The well established kinematic model for unicycle robot can be given in terms of its linear and angular speeds by

$$\dot{x} = u_1 cos\theta$$
$$\dot{y} = u_1 sin\theta \tag{4.1}$$
$$\dot{\theta} = u_2$$

where x and y represents the earth frame Cartesian coordinates of robot's center of gravity, θ is its angle with respect to the horizontal axis. u_1 and u_2 are linear and angular speed inputs, v and ω, through which we will control it. A unicycle UGV is illustrated in Fig. 4.1. u_1 and u_2 are directly connected with robot's right and left wheels' velocities u_R and u_L respectively. This relation is given by

$$u_1 = \tfrac{1}{2}(u_R + u_L)$$
$$u_2 = \tfrac{1}{2l}(u_R - u_L) \tag{4.2}$$
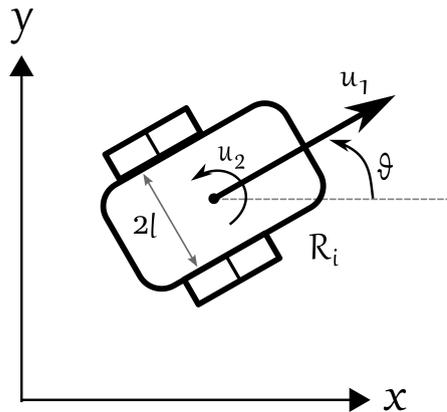
FIGURE 4.1: Variables of interest of a unicycle UGV

4.1.2 Control

As can be observed from eq. 4.2, the model has two inputs, namely u_1 and u_2 that we can work on to control three output states constructing the pose of it $[x \; y \; \theta]'$. This type of systems is known in control theory as underactuated system the control of which is considerably hard.

4.1.2.1 Virtual Inputs Hierarchical Control

In order for the nonholonomic UGV to track a desired trajectory $[X_d, \; Y_d]'$, we use a cascaded control scheme in which an outer loop control position states and produce desired orientation angle for the inner loop to control as depicted in Fig. 4.2. It decomposes the problem into two sub-problems: First, assuming holonomic dynamics track the trajectory. Then, since in practice dynamics does not allow for such behavior, calculate the necessary attitude of the system and ensure it is being followed.
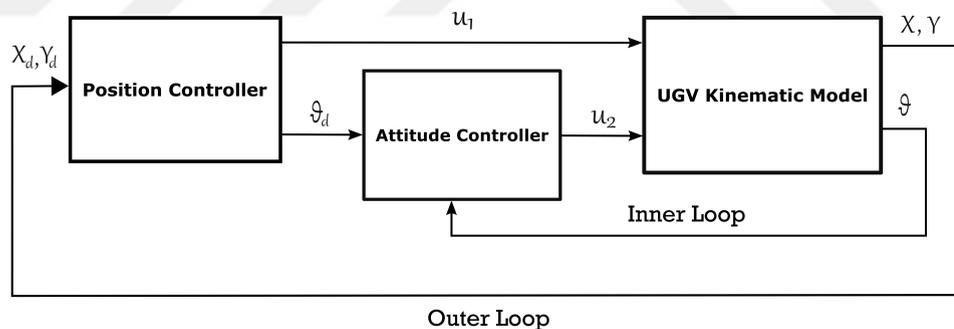
FIGURE 4.2: Hierarchical control scheme for nonholonomic UGV

Position Control

Position control bears the responsibility of generating the desired attitude desired values of the attitude , while following a feasible trajectory. The position is designed by using the first two equations of eq. 4.2. The virtual control inputs approach aims at forcing the errors, the difference between true and desired values, to converge to zero by means of cascaded PIDs. We first define position errors

and then differentiate to get error dynamics

$$e_x = x_d - x, \quad e_y = y_d - y$$
$$\dot{e}_x = \dot{x}_d - \dot{x}, \quad \dot{e}_y = \dot{y}_d - \dot{y} \tag{4.3}$$

When we equate our defined virtual inputs to the first derivatives of x and y, the following relations from error dynamics are obtained

$$\mu_x = \dot{x}_d - \dot{e}_x$$
$$\mu_y = \dot{y}_d - \dot{e}_y$$

We then define our virtual inputs by PID means to regulate errors and their derivatives by

$$\mu_x = \dot{x}_d + K_{p_x} e_x + K_{i_x} \int_0^t e_x dt + K_{d_x} \dot{e}_x$$
$$\mu_y = \dot{y}_d + K_{p_y} e_y + K_{i_y} \int_0^t e_y dt + K_{d_y} \dot{e}_y \tag{4.4}$$

Finally, from eq. 4.2 we have

$$\mu_x = u_1 cos\theta$$
$$\mu_y = u_1 sin\theta \tag{4.5}$$

With simple analytical manipulation, u_1 input and desired reference angle θ_d are found as follows

$$u_1 = \sqrt{\mu_x^2 + \mu_y^2}$$
$$\theta_d = \arctan\left(\frac{\mu_y}{\mu_x}\right) \tag{4.6}$$

Attitude Control

Attitude control regulates the orientation error to zero also using a PID -based virtual input for its angular velocity. We define error, error dynamics and virtual input for attitude as follows

$$e_\theta = \theta_d - \theta$$
$$\dot{e}_\theta = \dot{\theta}_d - \dot{\theta}$$
$$\mu_\theta = \dot{\theta}_d + K_{p_\theta} e_\theta + K_{i_\theta} \int_0^t e_\theta dt + K_{d_\theta} \dot{e}_\theta \qquad (4.7)$$

We need to find θ_d derivative to utilize it in attitude error dynamics. This can be done analytically by differentiating both sides of eq. 4.6 and the following relation is obtained

$$\dot{\theta}_d = \frac{\dot{\mu}_y \mu_x - \mu_y \dot{\mu}_x}{\mu_x^2 + \mu_y^2} \qquad (4.8)$$

Finally, μ_θ virtual input is equated to θ's derivative and the second input is fully defined by

$$u_2 = \mu_\theta \qquad (4.9)$$

With the proper tuning of gains, namely K_{p_x}, K_{i_x}, K_{d_x}, K_{p_y}, K_{i_y}, K_{d_y} K_{p_θ}, K_{i_θ} and K_{d_θ} nonholonomic mobile robot tracks its desired trajectory.

4.2   Quadrotor UAV

Quadrotor is a rotary-wing underactuated UAV. In the following section, the kinematics and dynamics of the quadrotor are provided. These models enable us to describe the motion of a quadrotor with respect to its inputs. Besides that, a synthesis using the virtual inputs concept is performed for the quadrotor UAV to track the trajectories produced by the 3D extended planner.
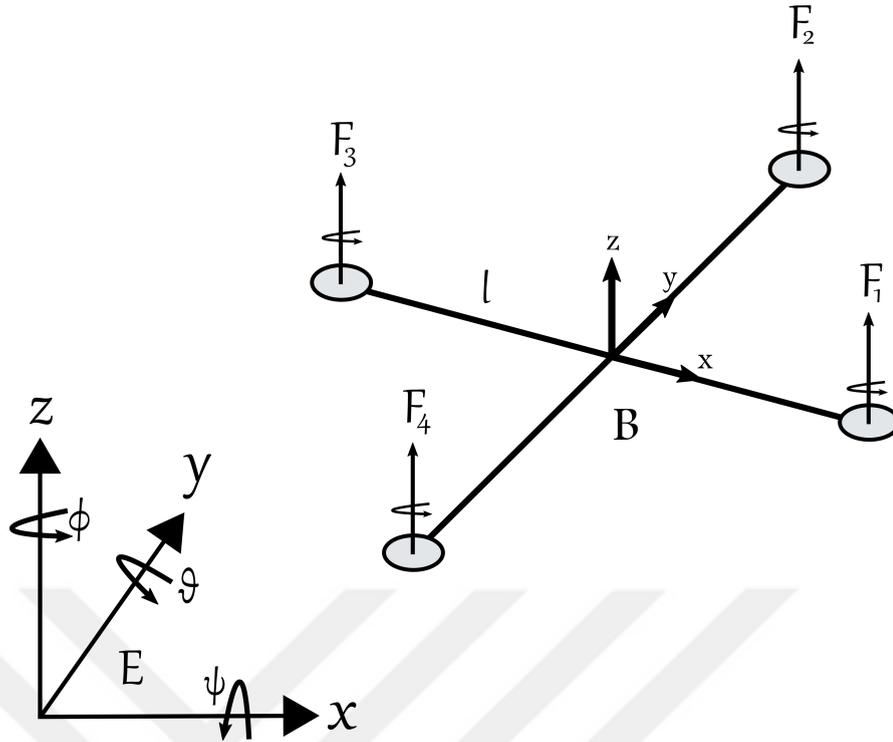
FIGURE 4.3: Coordinate systems for environment with a quadrotor UAV

4.2.1 Quadrotor's Kinematics and Dynamic Model

In order to describe kinematics and dynamics of a quadrotor we define two reference frames, the earth inertial frame (E frame) and body-fixed frame (B frame). Translational dynamics will be represented by a system of equations expressed in Earth frame, while attitude dynamics' system is expressed in Body frame. We use superscript to denote the frame with respect to which the parameter is defined. We define linear position Γ^E and angular position Θ^E of the quadrotor in Earth frame. While in body frame we have defined linear velocity v^B and angular velocity ω^B, rotor forces F^B and torques τ^B. Generalized position of the quadrotor is given as

$$\xi = \begin{bmatrix} \Gamma^E & \Theta^E \end{bmatrix}^T = \begin{bmatrix} X & Y & Z & \phi & \theta & \psi \end{bmatrix}^T \tag{4.10}$$

Generalized velocity is

$$v = \begin{bmatrix} V^B & \omega^B \end{bmatrix}^T = \begin{bmatrix} u & v & \omega & p & q & r \end{bmatrix}^T \tag{4.11}$$

The orientation of the body frame with respect to the earth frame is expressed with a rotation matrix R_Θ which is obtained by post-multiplying the three basic rotation matrices according to ZYX conversion

$$R_\Theta = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}$$

where $c_k = \cos k$, $s_k = \sin k$, and $t_k = \tan k$. We can relate derivative of a generalized position to the generalized velocity in the body frame as in eq. 4.12, where generalized J_Θ is composed of 4 sub-matrices

$$\dot{\xi} = J_\Theta v \tag{4.12}$$

$$J_\Theta = \begin{bmatrix} R_\Theta & 0_{3\times3} \\ 0_{3\times3} & T_\Theta \end{bmatrix} \tag{4.13}$$

T_Θ in eq. (4.13) refers to the transfer matrix defining the relation between the angular velocities in the E and those ones in the body-fixed frame B.

$$T_\Theta = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix}$$

The generalized velocity vector is

$$\zeta = \begin{bmatrix} \dot{\Gamma}^E & \omega^B \end{bmatrix}^T = \begin{bmatrix} \dot{X} & \dot{Y} & \dot{Z} & p & q & r \end{bmatrix}^T \tag{4.14}$$

The dynamics of the system in matrix form is

$$M_H \dot{\zeta} + C_H\left(\zeta\right)\zeta = G_H + O_H\left(\zeta\right)\Omega + E_H\left(\xi\right)\Omega^2 \tag{4.15}$$

where H is the hybrid frame from appending vectors from body frame and earth frame, $\dot{\zeta}$ is the quadrotor generalized acceleration vector, M_H is the system's inertia matrix, C_H is Coriolis-centripetal matrix, G_H is the gravitational vector

in H frame, $O_H(\zeta)$ is the gyroscopic propeller matrix. The third contribution considers the forces and torques directly produced by the main movement inputs.

We can rearrange eq. (4.15) to isolate the derivative of the generalized velocity vector $\dot{\zeta}$ with respect to the H frame

$$\dot{\zeta} = M_H^{-1}\left(-C_H(\zeta)\zeta + G_H + O_H(\zeta)\Omega + E_H(\xi)\Omega^2\right) \tag{4.16}$$

Finally, plugging the described matrices eq. (4.16) can be written as a hybrid system of equations

$$\begin{aligned}
\ddot{X} &= (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\frac{U_1}{m} \\
\ddot{Y} &= (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\frac{U_1}{m} \\
\ddot{Z} &= -g + (\cos\theta\cos\phi)\frac{U_1}{m}
\end{aligned} \tag{4.17}$$

$$\begin{aligned}
\dot{p} &= \frac{I_{YY}-I_{ZZ}}{I_{XX}}qr - \frac{J_{TP}}{I_{XX}}q\Omega + \frac{U_2}{I_{XX}} \\
\dot{q} &= \frac{I_{ZZ}-I_{XX}}{I_{YY}}pr - \frac{J_{TP}}{I_{YY}}p\Omega + \frac{U_3}{I_{YY}} \\
\dot{r} &= \frac{I_{XX}-I_{YY}}{I_{ZZ}}pq + \frac{U_4}{I_{ZZ}}
\end{aligned} \tag{4.18}$$

where

$$\begin{aligned}
\Omega_1^2 &= \frac{1}{4b}U_1 - \frac{1}{2bl}U_3 - \frac{1}{4d}U_4 \\
\Omega_2^2 &= \frac{1}{4b}U_1 - \frac{1}{2bl}U_2 + \frac{1}{4d}U_4 \\
\Omega_3^2 &= \frac{1}{4b}U_1 + \frac{1}{2bl}U_3 - \frac{1}{4d}U_4 \\
\Omega_4^2 &= \frac{1}{4b}U_1 + \frac{1}{2bl}U_2 + \frac{1}{4d}U_4
\end{aligned} \tag{4.19}$$

Eqs. 4.18 represents the translational dynamics which is underactuated since only one input for the system should drive three outputs states. Attitude dynamics, however, are represented by eqs. 4.17 and is fully actuated.

4.2.2 Control

The overall 6 DOF underactuated, highly nonlinear and coupled system's control is non-trivial. In this subsection, synthesis for quadrotors is briefly explained.

4.2.2.1 Virtual Inputs Hierarchical Control

Cascaded s for position and attitude are synthesized using virtual inputs approach. scheme is illustrated in Fig. 4.4.
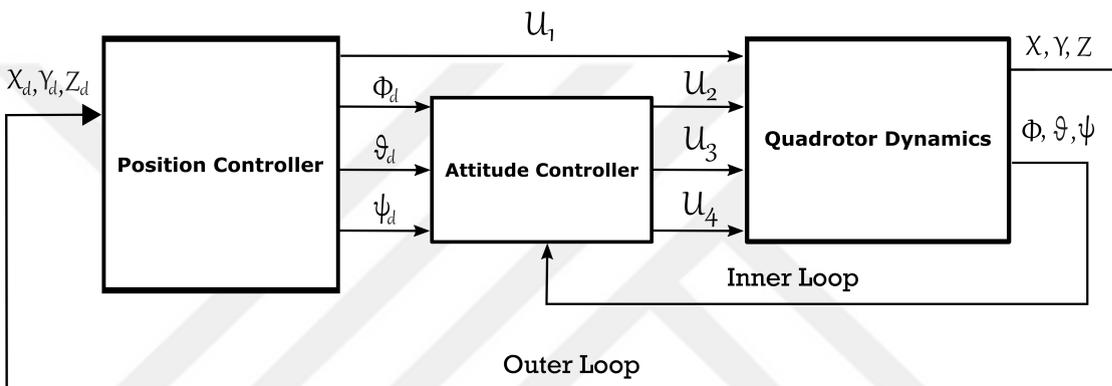
FIGURE 4.4: Hierarchical control scheme for quadrotor type UAV

Position Control

Position control block is a low-frequency outer loop control where it assumes being a holonomic ball that can go whenever it wants and plan the tracking trajectory accordingly. Outputs of this block are control input U_1 and desired angles that will make it possible for the previous assumption to hold.

Starting from translational dynamics (4.18) where U_1 is the control input, errors can be defined as

$$e_X = X_d - X$$
$$e_Y = Y_d - Y$$
$$e_Z = Z_d - Z \tag{4.20}$$

Then, error dynamics is found as

$$\dot{e}_X = \dot{X}_d - \dot{X} \Rightarrow \ddot{e}_X = \ddot{X}_d - \ddot{X}$$

$$\dot{e}_Y = \dot{Y}_d - \dot{Y} \Rightarrow \ddot{e}_Y = \ddot{Y}_d - \ddot{Y} \tag{4.21}$$

$$\dot{e}_Z = \dot{Z}_d - \dot{Z} \Rightarrow \ddot{e}_Z = \ddot{Z}_d - \ddot{Z} \tag{4.22}$$

Equating virtual inputs to position dynamics, we define

$$\ddot{X} = \mu_X$$

$$\ddot{Y} = \mu_Y$$

$$\ddot{Z} = \mu_Z \tag{4.23}$$

Thus, error dynamics in the Equation (4.21) becomes

$$\ddot{e}_X = \ddot{X}_d - \mu_X$$

$$\ddot{e}_Y = \ddot{Y}_d - \mu_Y$$

$$\ddot{e}_Z = \ddot{Z}_d - \mu_Z \tag{4.24}$$

Position control of the vehicle is reduced to the control of a double integrator through the following virtual controls:

$$\mu_X = \ddot{X}_d + K_{p,X} e_X + K_{d,X} \dot{e}_X + K_{i,X} \int e_X dt$$

$$\mu_Y = \ddot{Y}_d + K_{p,Y} e_Y + K_{d,Y} \dot{e}_Y + K_{i,Y} \int e_Y dt$$

$$\mu_Z = \ddot{Z}_d + K_{p,Z} e_Z + K_{d,Z} \dot{e}_Z + K_{i,Z} \int e_Z d \tag{4.25}$$

Using simple trigonometric relations the virtual controls are transformed to desired roll ϕ_d and pitch angles θ_d.

$$\phi_d = \arcsin\left(\frac{\sin(\psi_d)\,\mu_X - \cos(\psi_d)\,\mu_Y}{\sqrt{\mu_X^2 + \mu_Y^2 + (\mu_Z + g)^2}}\right)$$

$$\theta_d = \arcsin\left(\frac{\cos(\psi_d)\,\mu_X + \sin(\psi_d)\,\mu_Y}{\cos(\phi_d)\sqrt{\mu_X^2 + \mu_Y^2 + (\mu_Z + g)^2}}\right) \tag{4.26}$$

Yaw angle ψ is assumed to be some fixed value ψ_d, and total thrust in terms of the virtual inputs is given by

$$U_1 = m\sqrt{\mu_X^2 + \mu_Y^2 + (\mu_Z + g)^2}$$

Attitude Control

Attitude control is a high-frequency inner loop that works towards fulfilling the high-level position 's requirements of roll ϕ, pitch θ (yaw angle is fixed $\psi_d = \psi$) such that appropriate torque signals responsible for steering the quadrotor in the desired direction are synthesized.

Errors in attitude angles can be defined as

$$e_\phi = \phi_d - \phi$$

$$e_\theta = \theta_d - \theta$$

$$e_\psi = \psi_d - \psi \tag{4.27}$$

Virtual inputs for the angular position are designed as PIDs therefore we get the equation for roll control

$$\ddot{\phi} = \ddot{\phi}_d + K_{p,\phi} e_\phi + K_{d,\phi} \dot{e}_\phi + K_{i,\phi} \int e_\phi dt \tag{4.28}$$

where the control input U_2 is designed as

$$U_2 = I_{XX} \left(\ddot{\phi}_d + K_{p,\phi} e_\phi + K_{d,\phi} \dot{e}_\phi + K_{i,\phi} \int e_\phi dt \right) \tag{4.29}$$

In a similar way we obtain the expression for U_3 and U_4

$$U_3 = I_{YY} \left(\ddot{\theta}_d + K_{p,\theta} e_\theta + K_{d,\theta} \dot{e}_\theta + K_{i,\theta} \int e_\theta dt \right)$$

$$U_4 = I_{ZZ} \left(\ddot{\psi}_d + K_{p,\psi} e_\psi + K_{d,\psi} \dot{e}_\psi + K_{i,\psi} \int e_\psi dt \right) \tag{4.30}$$

where K_p, K_d and K_i are proportional, derivative and integral controller gains respectively.

Chapter 5

Collision Avoidance Framework via Virtual Shells

Collision avoidance is one of the central problems in the coordinated motion of a group of autonomous mobile robots. As discussed in chapter 2, many decentralized coordination models including APFs and virtual spring-damper bonds, although mathematically sound, do not guarantee collision-free paths and require either scenario-dependent offline parameter tuning or the aid of computationally costly algorithms. The problem gets even more complicated when kinematic and dynamic constraints of the robots are taken into account. Nonholonomic robots, for example, cannot arbitrarily change their orientations, thus decreasing their probability of maneuvering a predicted collision. In this chapter, an intuitive framework, inspired by fish schools in nature, to detect and avoid collisions online is proposed. Although the algorithm is extendable to environment's static/dynamic obstacle avoidance, we restrict our discussion to inter-robot collisions as a performance criterion.

The newly proposed collision avoidance framework eliminates geometry complexities of robots of different sizes, shapes or dynamics. Moreover, it decomposes the problem into 2 sub-problems:

- Collision Detection.
- Collision Response.

Figure 5.1 shows the proposed framework integrated into the planner block, hence enhancing the overall scheme by successfully eliminating collisions during coordination.

HIGH LEVEL (Planner)　　　　　　　LOW LEVEL (Control)

Reference Trajectory Generation
- Decentralized Configuration
- Virtual Point Mass Model
- Virtual Spring-Damper Layers

Physical Robots Motion Control
- Actual Dynamical Models
- Reference Tracking Control
- Fetch Poses for Feedback

DESIRED TRAJECTORIES

VELOCITY UPDATE

Inter-Robot Collision Avoidance
- Virtual Shell Abstraction
- Virtual Shell Coll. Detection
- Collision Resolution/Response

STATE FEEDBACK PATH

FIGURE 5.1: Enhanced hierarchical coordination framework

5.1 Virtual Shells

Virtual Shells concept is inspired by the allowable smooth and harmless collisions in some types of animal swarms in nature e.g. fish schools Fig. 5.2(a), where the small scale, flexible bodies and force damping medium reduces the risk to zero. Instead of the common concept of "preventing collisions", we propose the concept of "allowable safe collisions" as in bumper cars Fig. 5.2(b). It is based on transferring collisions from the physical robots level, which might cause hardware damage and task failure, to the level of virtually synthesized shells with predefined geometry enveloping the robots.

After a collision is detected, an action/response necessary to escape collision is triggered. This response is a reference trajectory modification through a velocity

(a)

(b)
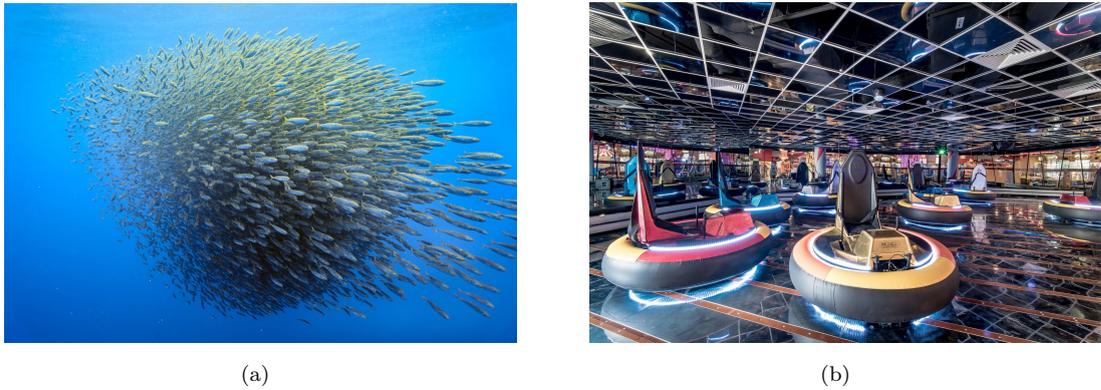
FIGURE 5.2: (a) school of fish in oceans, (b) bumper cars in amusement parks

update according to some law. The low-level controller bears the responsibility for aligning the core physical robot at the center of the moving shell.

This framework is extendable and can be implemented in both 2D and 3D multi-robot environments. Figure 5.3 illustrates the concept of virtual shells for different sample robots.

(a)

(b)

FIGURE 5.3: Virtual shells enveloping heterogeneous robots: (a) 2d environment case and (b) 3d environment case.

There are three main considerations to note regarding the proposed framework, most of which are affected by perception and communication constraints of the system. They can be summarized as:

- Shell Geometry
- Collision Detection
- Collision Response using Velocity Update Laws

5.2 Shell Geometry

Choosing the virtual shell's geometry is crucial in the sense that it directly affects the collision detection law. Simpli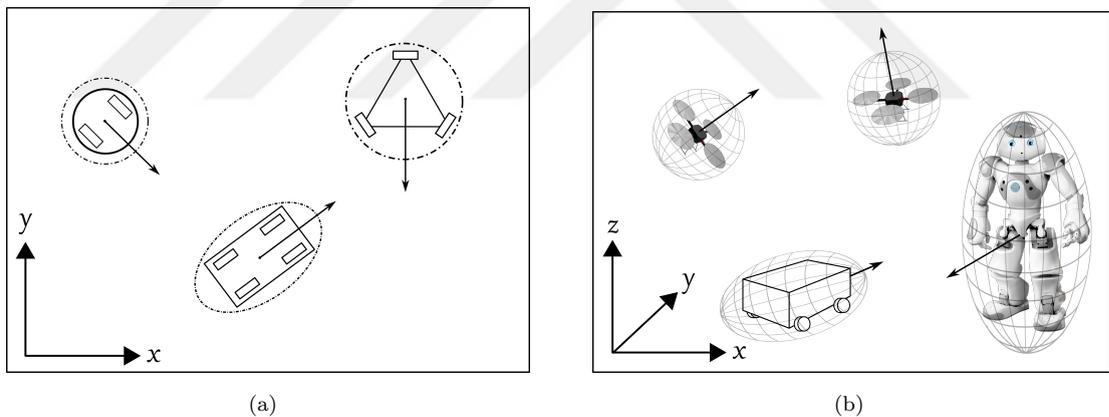fying complex robots' geometries is one of this framework's objectives. Therefore, highly symmetric shell geometry is undoubtedly preferred. Figure 5.4 suggests two possible shell geometries based on the aforementioned criteria, namely a circle and an ellipse. In 3D workspaces, these suggested shapes become a sphere and an ellipsoid.

(a)   (b)
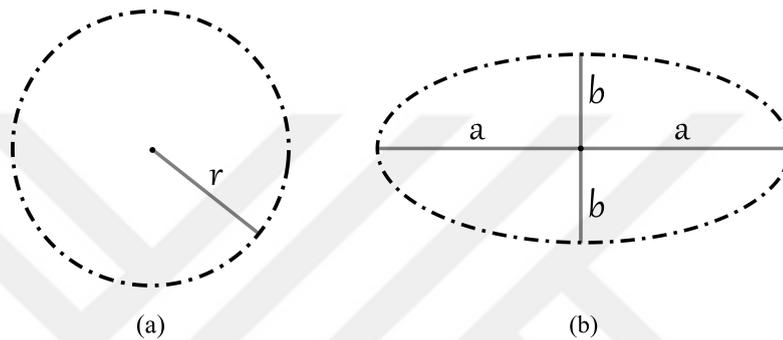
FIGURE 5.4: Highly symmetric shell examples in 2d: (a) circle and (b) ellipse

As a proof of concept, throughout the rest of this thesis, we restrict our discussion of shell geometry to the highly symmetric circular form (spherical in 3D). We define a circular virtual shell $\Omega_i(r)$ of radius r, enveloping robot R_i centered at X_i and having a velocity vector V_i as depicted in figure 5.5.

FIGURE 5.5: Virtual Shell $\Omega_i(r)$, for the robot, R_i

Radius r of the shell is predefined offline, and can be assigned taking many factors into consideration including:

- Environment geometrical constraints
- Average operating speed

From which one can infer and synthesize a simple radius assigning law. In our simulations we arbitrarily choose r for experimentation, granted that it satisfies $r \leq \beta L$, where β is a safety factor and L is the radius of the circumscribed circle of the robot.

5.3 Collision Detection

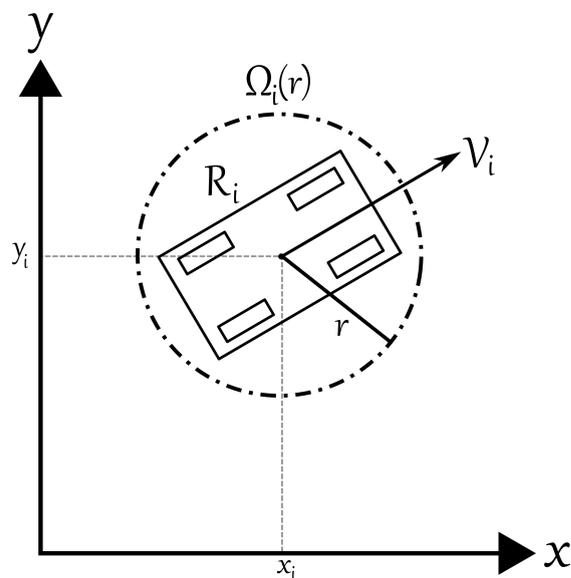Unlike a large number of algorithms in the literature based on beforehand robot-level collision "prediction" [54],[64],[65], the proposed framework shifts the problem to virtual shells-level collision "detection". This is made possible due to the safety offset inherent to the virtual shell.

For a circular virtual shell $\Omega_i(r_i)$ enveloping robot R_i, a collision is detected with another circular virtual shell $\Omega_j(r_j)$ enveloping robot R_j in the cases of contact or overlapping between the shell-pair, i.e., the distance between their centers is smaller or equal to the sum of their radii as depicted in figure 5.6.
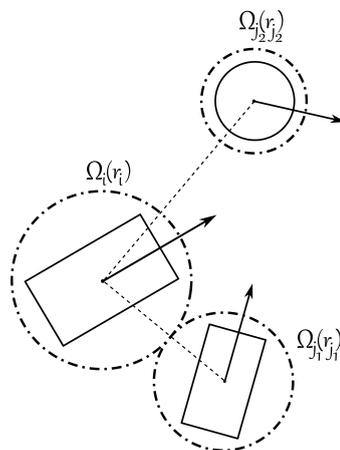

FIGURE 5.6: Collision of $\Omega_i(r_i)$ with $\Omega_j(r_j)$ is detected due to contact

A flag is triggered whenever $\|X_i - X_{j_k}\| \leq (r_i + r_{j_k})$ is satisfied indicating which shell-pair is colliding. We assume single collision at a time for the sake of simplicity,

in case of simultaneous multiple collisions, we give response priority according to the magnitude of the collision.

The case becomes more complex when considering ellipse/ellipsoid shaped shells or a combination of them. Nevertheless, necessary conditions for such cases can be derived analytically and built upon for a more efficient collision avoidance. This is true as it can compensate for the nonholonomic constraint, something that circular shell cannot do. This is considered to be a future work in this thesis.

Algorithm 5.1 Circular/Spherical Shell Collision Detection

Require: Robot R_i perceives its n nearest neighbors R_{j_1}, $R_{j_2} \dots R_{j_n}$
  procedure COLLISION DETECTION(r_i,$r_{j_1} \dots r_{j_n}$)
    Initialize $Flag$ array
    Calculate $\|X_i - X_{j_1}\| \dots \|X_i - X_{j_n}\|$
    for k from 1 to n **do**
      if $\|X_i - X_{j_k}\| \leq (r_i + r_{j_k})$ **then**
        $Flag(k) \leftarrow true$
      else
        $Flag(k) \leftarrow false$
      end if
    end for
    Return $Flag$
  end procedure

---

where $(X_i, X_{j_1} \dots X_{j_n})$ are robots' position vectors in earth frame E and $(r_i, r_{j_1} \dots r_{j_n})$ are their corresponding shells' radii respectively.

5.4   Collision Response

After a collision flag $flag(k)$ is triggered true, the robot of study, R_i, needs to act based on an algorithm hence, safely avoiding the collision with R_{j_k} not sabotaging the main task of coordinated motion. Possible responses in the literature are discussed in chapter 2. We hereby propose two "physical phenomena inspired" online algorithms that modify reference trajectory's velocity profile, hence producing an improved collision-free reference trajectory for the low level controller to track.

5.4.1 Elastic Collision (EC) Algorithm

Elastic collision of two rigid bodies is a well-formulated physical phenomenon where the involved objects separate after collision with no deformations or energy loss, e.g., billiard balls. The system of colliding objects form an isolated system and both total kinetic energy and total linear momentum are conserved, i.e. the coefficient of restitution will equal *one*.
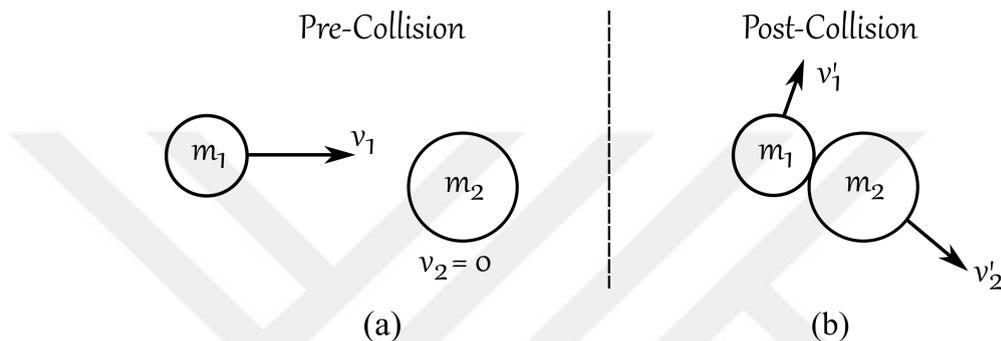
FIGURE 5.7: Elastic collision of two rigid bodies in 2d: (a) before collision and (b) after collision

We model the collision between two virtual shells $\Omega_i(r_i)$ and $\Omega_j(r_j)$ as an elastic collision between two rigid bodies of masses m_i and m_j corresponding to their core robots. Post-collision velocity law is used the necessary and sufficient action to modify robot's reference trajectory generated by the planner online, thus avoiding the collision.

5.4.1.1 Post-Collision Velocity Law Derivation

For two colliding shells $\Omega_i(r_i)$ and $\Omega_j(r_j)$ with center position vectors X_i, X_j and velocity vectors $\dot{X}_i, \dot{X}_j$ respectively, we review the derivation for 1D, 2D and 3D cases as follows:

1D case

This case is a backbone for higher order case derivations, since the velocity change in an elastic collision is restricted to its components on the radial axis, i.e., the one connecting the centers of the two rigid bodies passing through the collision point. In this case, center position and velocity vectors become scalars, thus simplifying the algebra for derivation:

Conservation of momentum: $\quad m_i\dot{X}_i + m_j\dot{X}_j = m_i\dot{X}'_i + m_j\dot{X}'_j$

Conservation of kinetic energy: $\frac{1}{2}m_i\dot{X}_i^2 + \frac{1}{2}m_j\dot{X}_j^2 = \frac{1}{2}m_i\dot{X}'^2_i + \frac{1}{2}m_j\dot{X}'^2_j$

Combining these two equations and performing some algebraic manipulations we get post-collision velocities for circular shell $\Omega_i(r_i)$ as follows:

$$\dot{X}'_i = \frac{\dot{X}_i(m_i - m_j) + 2m_j\dot{X}_j}{m_i + m_j} \tag{5.1}$$

where m_i, m_j are the masses of the robots R_i, R_j positioned at the centers of the colliding virtual shells $\Omega_i(r_i)$ and $\Omega_j(r_j)$ respectively.
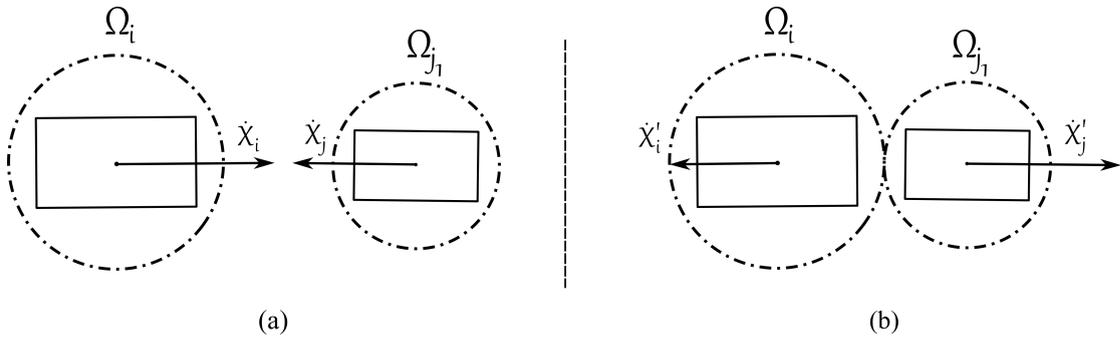
FIGURE 5.8: Virtual shells elastic collision model (1D): (a) before collision and (b) after collision

There is no need to calculate $\dot{X}'_j$ as every robot in our decentralized system is considered to be R_i from its perspective. By the same token, its two perceived nearest neighbors are considered to be R_{j_1} and R_{j_2}.

When $m_i = m_j$, eq. 5.1 becomes $\dot{X}'_i = \dot{X}_j$ indicating that for the collision of similar masses objects (shells) velocities after collision are simply exchanged.

2D case

In this case, as already mentioned, tangential component of the velocities remains unaltered while radial component can be found using the one-dimensional collision formula:

$$\dot{X}'_{ir} = \frac{\dot{X}_{ir}(m_i - m_j) + 2m_j\dot{X}_{jr}}{m_i + m_j} \quad , \quad \dot{X}'_{it} = \dot{X}_{it} \tag{5.2}$$

We then return the scalar values of radial and tangential velocities into their vector:

$$\vec{\dot{X}}'_{ir} = \dot{X}'_{ir}.\vec{U}_r \quad , \quad \vec{\dot{X}}'_{it} = \dot{X}'_{it}.\vec{U}_t \tag{5.3}$$

Then we finally reconstruct the complete two dimensional velocity vector in earth frame E by adding the radial and tangential components:

$$\vec{\dot{X}}'_i = \vec{\dot{X}}'_{ir} + \vec{\dot{X}}'_{it} \tag{5.4}$$

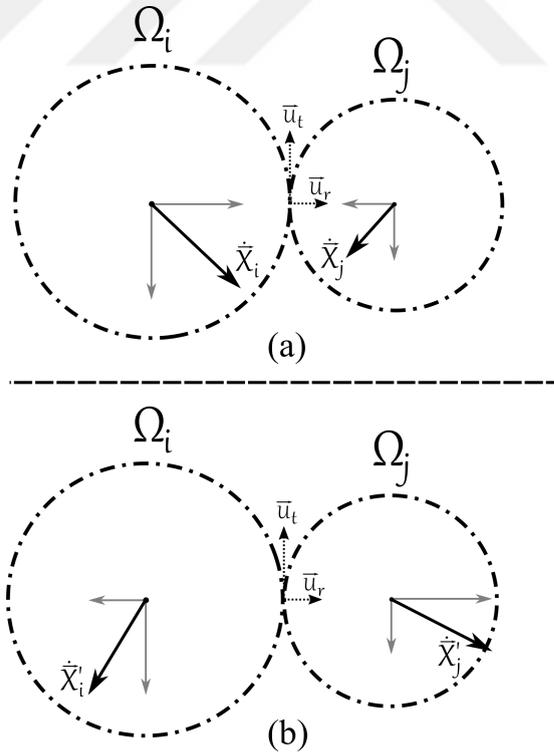where $\vec{U}_r$ and $\vec{U}_t$ are the radial axis and tangential axis unit vectors respectively.

FIGURE 5.9: Virtual shells elastic collision model (2D): (a) before collision and (b) after collision

3D case

This case is a further extension from 1D case in a similar manner to 2D case derivation. Consequently, velocity update laws are driven exactly the same way as in 2D case. The difference is that we will have two perpendicular tangential axes where velocity components remains unchanged instead of one, as depicted in figure 5.10.
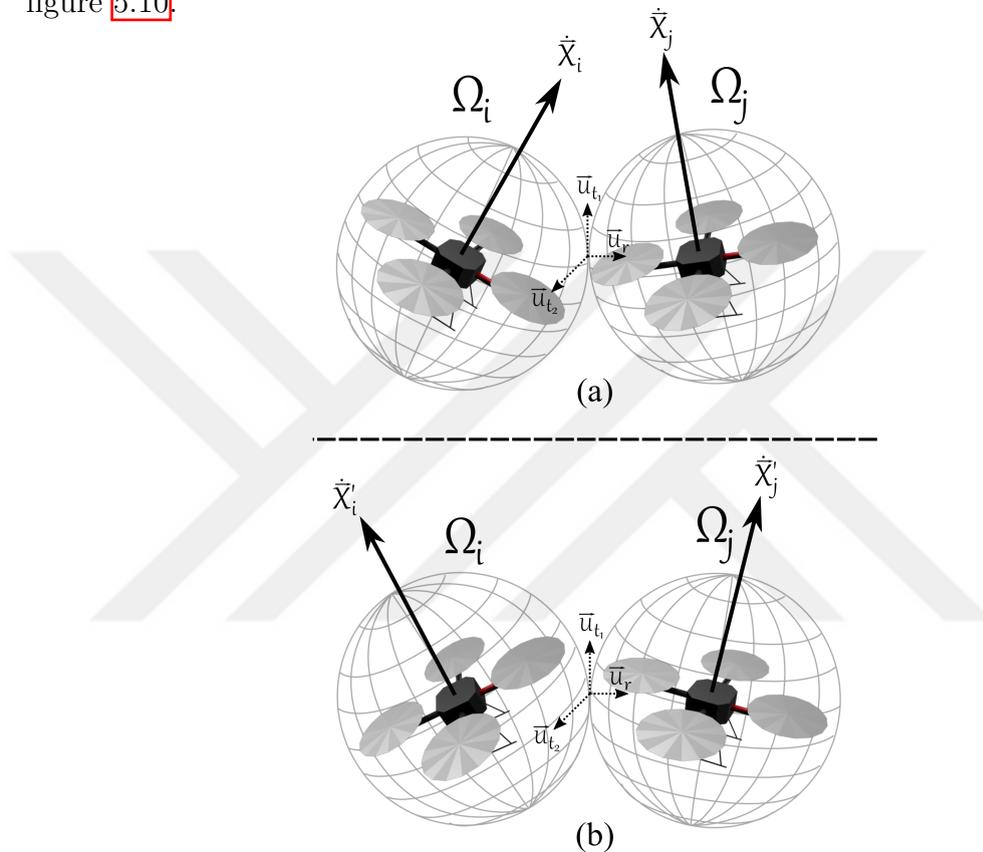
FIGURE 5.10: Virtual shells elastic collision model (3D): (a) before collision and (b) after collision

5.4.2 Light Beam Reflection (LBR) Algorithm

The elastic collision model assumes a high-level of communication in-between the colliding robots in order to get the velocity information of the other robot. This is consistent with the general coordination scheme as virtual spring-damper bonds also require such information. However, this might be difficult to implement in practice. One possible solution is to add a planner agent that have global information, yet this will decrease the system's robustness.

Perception based solutions, like observers, are also valid. Where robot can estimate the velocity of the robot due to which a collision was detected. However, estimation accuracy and hardware cost are considerable issues. We propose an alternative abstract algorithm inspired by the light beam reflection physical phenomenon where the light is considered as a ball hitting a rigid wall of infinite mass. This algorithm does not necessitate the perception and communication requirements that the aforementioned algorithm do.

When R_i detect a collision due to R_j, R_i is considered to hit a wall W_j with an inclination coincident to the tangential axis of collision. The change after collision is simply a reflection in direction of the radial component of R_i robot's velocity. The velocity of the other robot is not taken into consideration as discussed earlier:

$$\dot{X}'_{in} = -\mu.\dot{X}_{in} \qquad , \qquad \dot{X}'_{it} = \dot{X}_{it} \tag{5.5}$$

where μ is a pre-tuned magnitude scale factor by the planner. The final velocity vector $\dot{\vec{X}}'_i$ is found in a similar steps as in eqs. 5.3 and 5.4. An illustrative two dimensional scenario is depicted in figure 5.11
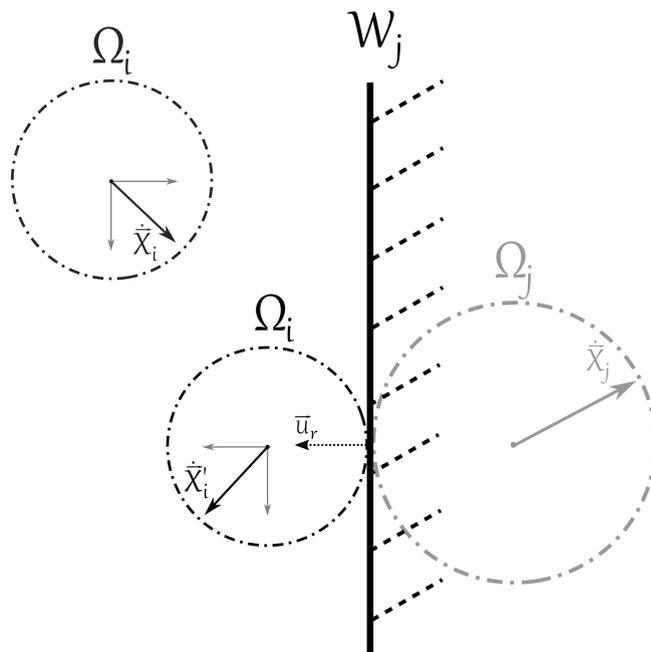

FIGURE 5.11: Virtual shells light beam reflection model (2D)

A disadvantage of this algorithm is being relatively less efficient, e.g., a head to tail collision scenario will force the robot to turn back which will affect the overall coordinated motion due to existing virtual spring-damper connections.

	Algorithm1	**Algorithm2**
Communication Requirements	Intensive	Minimal
Computational Complexity	Relatively Complex	Simple
Execution Time	Shorter	Longer
Task Fitness	Better Fit	Inefficient

TABLE 5.1: A comparison between elastic collision and light beam reflection algorithms

5.4.3 General Velocity Update Algorithm

Since collision response is separated from collision detection, other algorithms can be utilized in the future. The newly obtained velocity is applied to R_i's reference trajectory's velocity for a T_{hold} time until the flag is no longer triggered hence the collision is avoided. Regardless of the selected method, the proposed process is now as follows:

Algorithm 5.2 Collision Response: Post-Collision Velocity Update

---
Require: COLLISION DETECTION algorithm implemented
  procedure COLLISION RESPONSE($Flag, \dot{\vec{X}_i} \ldots \dot{\vec{X}_{jn}}$)
    for k from 1 to n **do**
      if $Flag(k) = 1$ **then**   ▷ if Collision with robot R_{j_k} is detected
        for T_{hold} period of time **do**   ▷ hold post-collision velocity for T_{hold}
          $\dot{\vec{X}_i} \leftarrow \dot{\vec{X}_i}'$
        end for
      else   ▷ if no Collision was detected
        $\dot{\vec{X}_i} \leftarrow \dot{\vec{X}_i}$ ▷ do not change reference velocity profile
      end if
    end for
    return $\dot{\vec{X}_i}$
  end procedure

The algorithm assumes that no more than one collision is happening at the same sample time. Successive collisions can happen, and it is the planners choice to choose T_{hold} accordingly.

Chapter 6

Simulation Results and Discussion

Computer simulations and animations were carried out to test the performance of the developed scheme; both for "coordinated motion planning and control" part of chapters 3, 4 and for "collision avoidance framework" part of chapter 5.

All simulations were performed in Matlab/Simulink environment. In these simulations, an MRS was considered for a coordinated task defined by a circular formation of its robots around the goal point. The framework is tested in 2D environments using groups of three, four, five and ten nonholonomic UGVs. Additionally, it is tested in 3D environments using groups of three and five quadrotor UAVs. Several functionally distinct scenarios were first established, then implemented and followed by a brief discussion.

While performing simulations, controller errors and environment disturbances are crucial factors that need to be compensated for. This particularly valid for collision detection sub-problem as the controller is never ideal and cannot 100% follow the reference trajectory thus causing lead/lag in the response. We overcome these factors by routing our feedback path from the actual poses instead of the reference ones as depicted in Fig. 6.1. The essential problem of inter-robot collisions avoidance is, therefore, solved in an online manner.
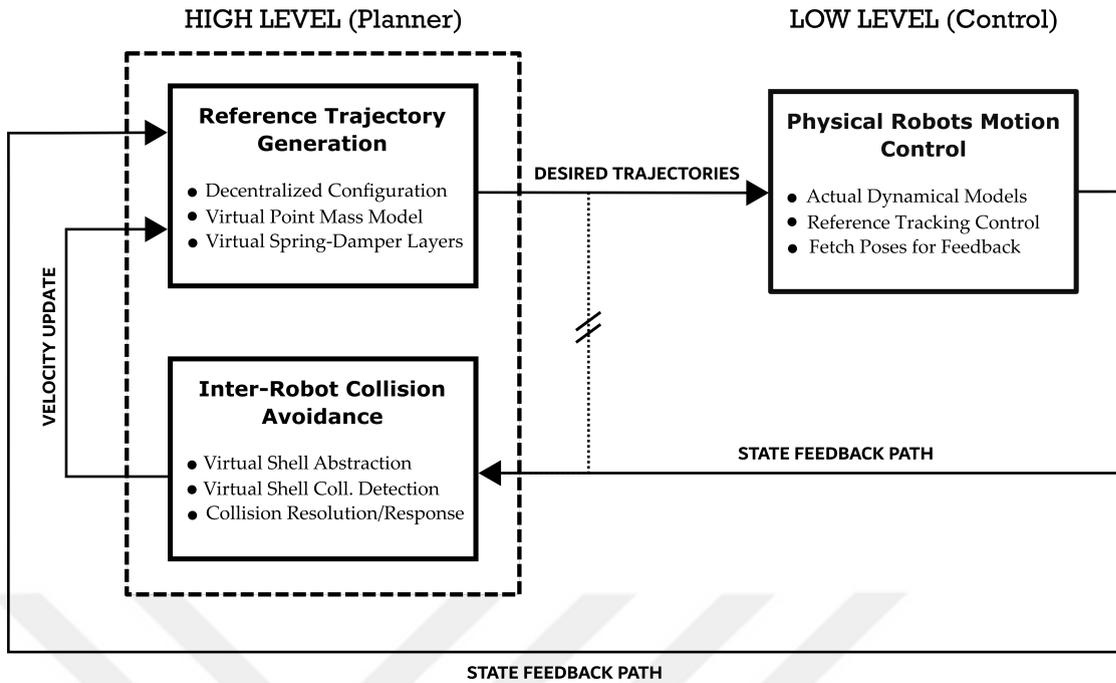
FIGURE 6.1: Enhanced hierarchical coordination framework block diagram

6.1 2D Workspace implementation

The virtual masses used in simulations are all set as $m_i = 1$ $[kg]$. Physical nonholonomic UGV robots' dimensions are $64 * 32$ $[mm]$ while their masses are equal to those of the virtual masses used for reference trajectory generation. Virtual shells $\Omega_i(r_i)$s' radii were set to $r_i = 55$ $[mm]$ according to a safety factor of $\beta \simeq 1.5$. T_{Hold} is set to a small value of 0.05 $[sec]$. Controller values are fixed to $k_p = 23$, $k_i = 3$ and $k_d = 0$, as a result of gain adjustment process on a single non holonomic robot.

Some system's parameters are showed in Tab. 6.1. Their suitable values were determined experimentally according to the performance of successive simulations on a single quadrotor UAV. They will be hold fixed during the following simulation scenarios unless otherwise indicated. Some other parameters, however, will be scenario-dependent and tabulated separately.

Parameter	Value	Reference Equation
b	1	3.1
k_R	16	3.3
c_R	5	3.3
k_G	15	3.4
c_G	10	3.4
$k_R{}'$	9	3.6
α	2	3.6
γ	1	3.6

TABLE 6.1: Simulation parameters

6.1.1 Coordinated Motion Simulations

We first simulate an MRS of three robots, the basic module in the system, illustrating main operational behaviors of the reference trajectory planner, collision avoidance framework and designed controllers. Then, it is followed by simulations on groups of four and five robots where the modular design is verified to some extent. Finally, in order to inspect the overall structure's potential and collision avoidance model's robustness, we simulate a group of ten nonholonomic robots in two realistic *swarm* scenarios.

In each scenario, important frames from the coordinated motion animation are emphasized, e.g. collision detected, collision avoided, formation obtained ... The **x** mark indicates the goal point G while the black dashed circle around it is d_G circle, where the robots are supposed to uniformly form around G. The most outer cyan colored circle is d_{break} circle, where k_R smoothly transform into $d_R{}'$.

Robots are originally distinguished by their shell's and orientation vector's color. When a robot R_i detects a collision due to another robot R_j, its face color changes to R_j's color. We first implement elastic collision model and then repeat the simulation with the alternative collision response model of light beam reflection. For the sake of compactness, we compare and comment on their performance in detail under scenario 2 and 3, and then select one to use in the following next scenarios. Controllers performance is also verified in the form of desired vs. controlled state graphs.

Scenario 1

In this scenario, the operation of the coordination scheme without triggering the collision avoidance framework is demonstrated. The robots are initially placed inline with sparse distances away from the goal where $X_1 = [-4 \ \ 4]'$, $X_2 = [0 \ \ 4]'$ and $X_3 = [4 \ \ 4]'$. Conservative values for some parameters were deliberately assigned to avoid collision situation, e.g. large approaching inter-robot distances with rigid enough virtual damper to escape from collision scenarios. Scenario-dependent parameters of the simulation are mentioned in Tab. 6.2.

Parameter	Value
G	$(0 \ - 2)$
d_G	1.5
d_{break}	2
d_R	2
c_R	7

TABLE 6.2: Scenario-1: modified parameters

As it can be observed from animation's snapshots in Fig. 6.2, robots first approach each other and move in a coordinated fashion towards G. Once they've passed d_{break} circle, they spread around the d_G circle in a preparation for achieving the desired uniform formation around G.

Scenario 2

The simulation was run for a group of three robots placed around the goal with one robot closer than the other two robots $X_1 = [0 \ \ 4]'$, $X_2 = [-4 \ \ -4]'$, $X_3 = [4 \ \ -4]'$ and $G = [0 \ \ -1]'$. We demonstrate a single head-to-head collision case to test the proposed framework. Fig. 6.3 represents snapshots from the animation done using EC model, while Fig. 6.4 is constructed using LBR model. In either case, the collision is first detected and then successfully avoided without sabotaging the overall mission.
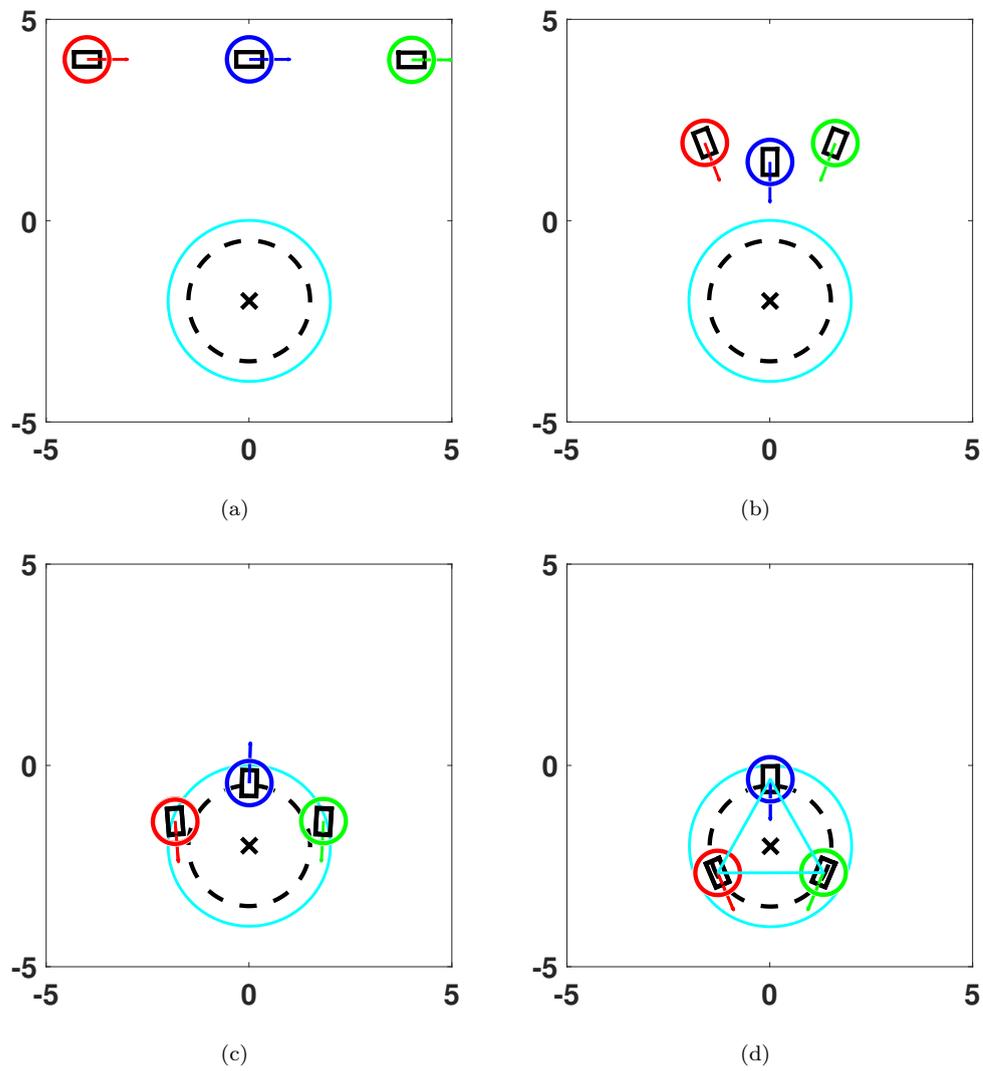
FIGURE 6.2: Scenario-1: (a) initial configuration, (b) coordinated motion, (c) spreading out after d_{break} and (d) uniform formation

EC model based avoidance, Fig. 6.3, is more *aware* to its core robot's mass, velocity vector and similarly colliding robot's mass and velocity vector. Thus, more reasonable behavior was expected on the expense of communication requirements.

Since colliding robots have identical masses, the post-collision velocity values reduce from eq. 5.1 to simply an exchange radial components of their velocity vectors eventually escaping collision status .
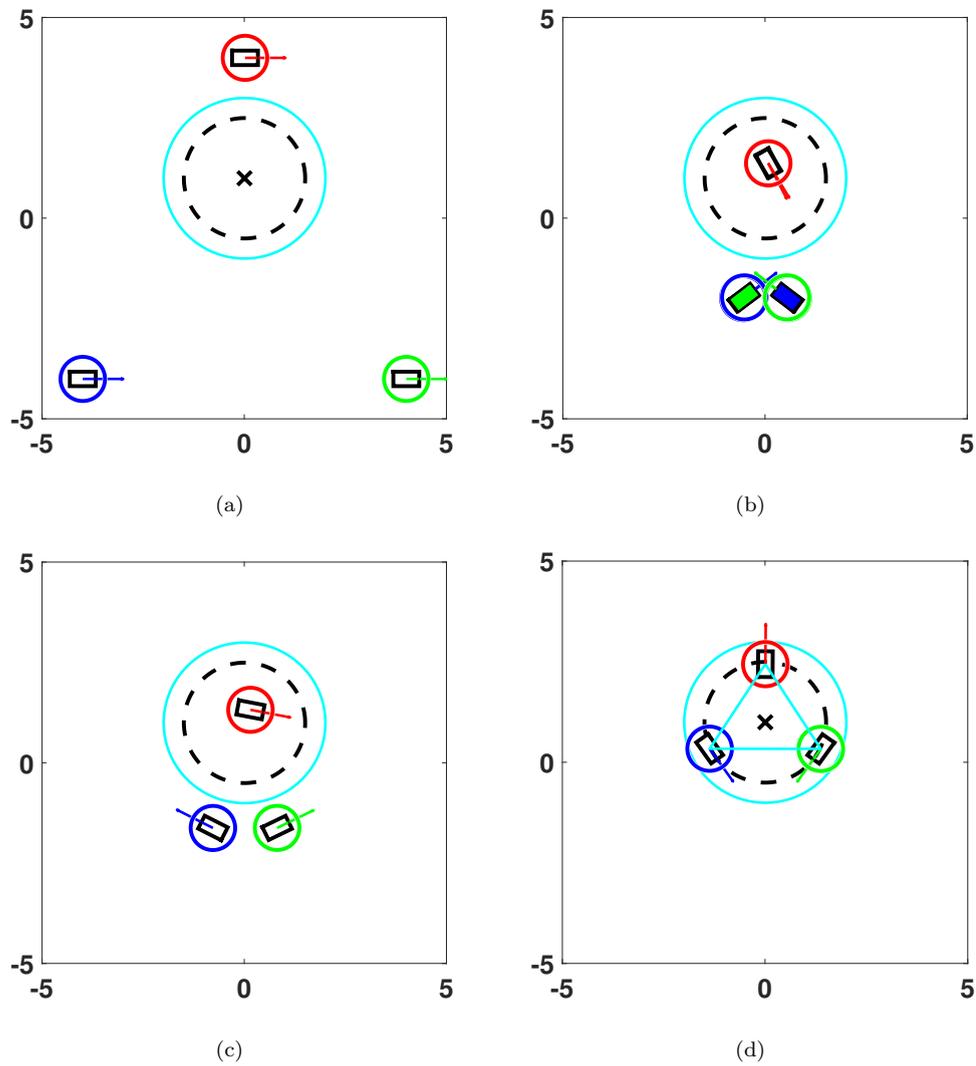
FIGURE 6.3: Scenario-2E: (a) initial config., (b) h-to-h collision detection, (c) collision avoidance (EC model) and (d) uniform formation

With LBR model, Fig. 6.3, longer convergence time is needed as obvious in the controllers' state graphs, Fig. 6.9. This is due to the unnecessary large bounce from each robot at the collision. Nevertheless, the coordinated task was achieved and the formation was constructed with the only difference being the execution time. As a conclusion, in similar head-to-head collision scenarios, the difference between the two models is evaluated to be tolerable.
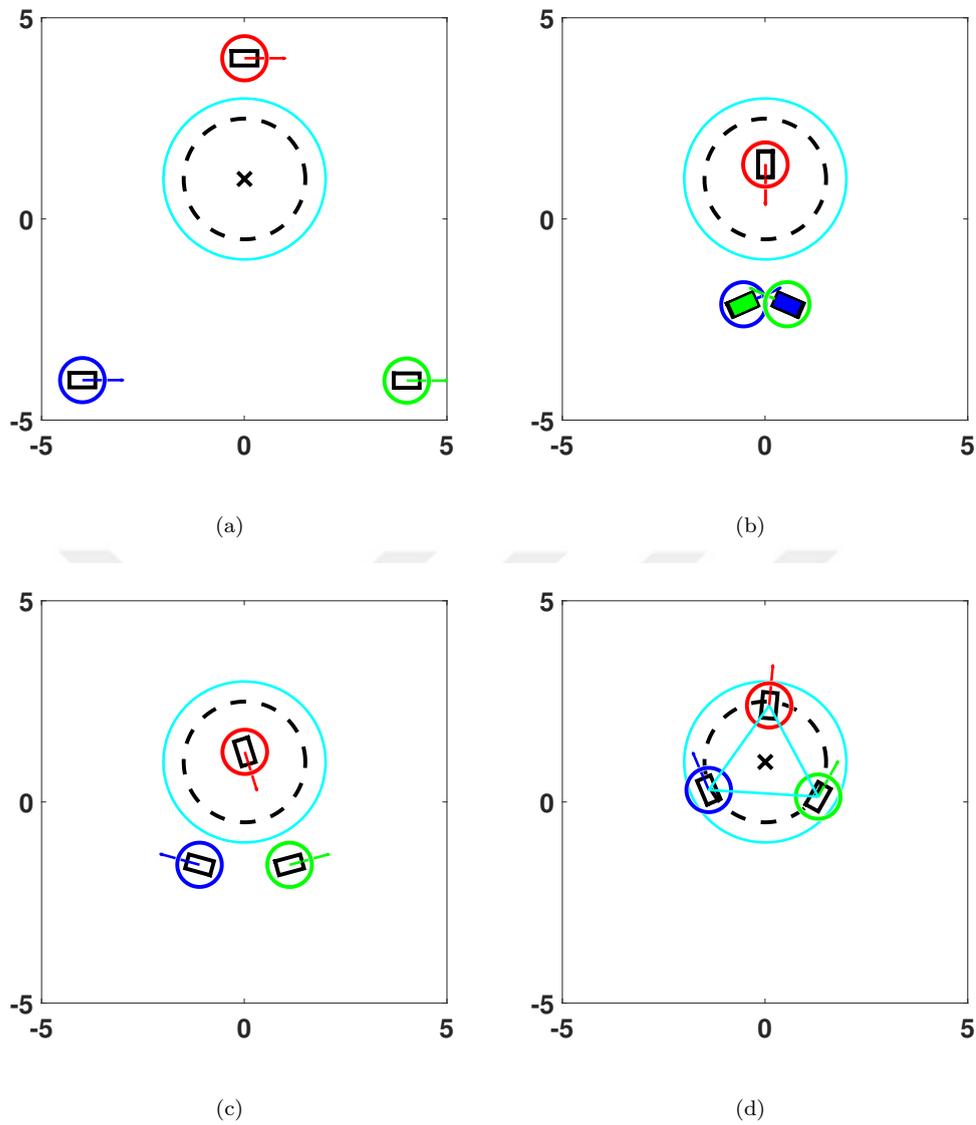
FIGURE 6.4: Scenario-2R: (a) initial config., (b) h-to-h collision detection, (c) collision avoidance (LBR model) and (d) uniform formation

Scenario 3

A head-to-tail collision scenario is investigated to further compare the two response algorithms. We place the robots and the goal on the main diagonal of the workspace: $X_1 = [-4 \ 4]'$, $X_2 = [-1 \ 1]'$, $X_3 = [4 \ -4]'$ and $G = [1.5 \ -1.5]'$. For demonstration purposes, we relax c_R to 1. The two robots on the same side with respect to goal, mimic a simple platooning motion situation.
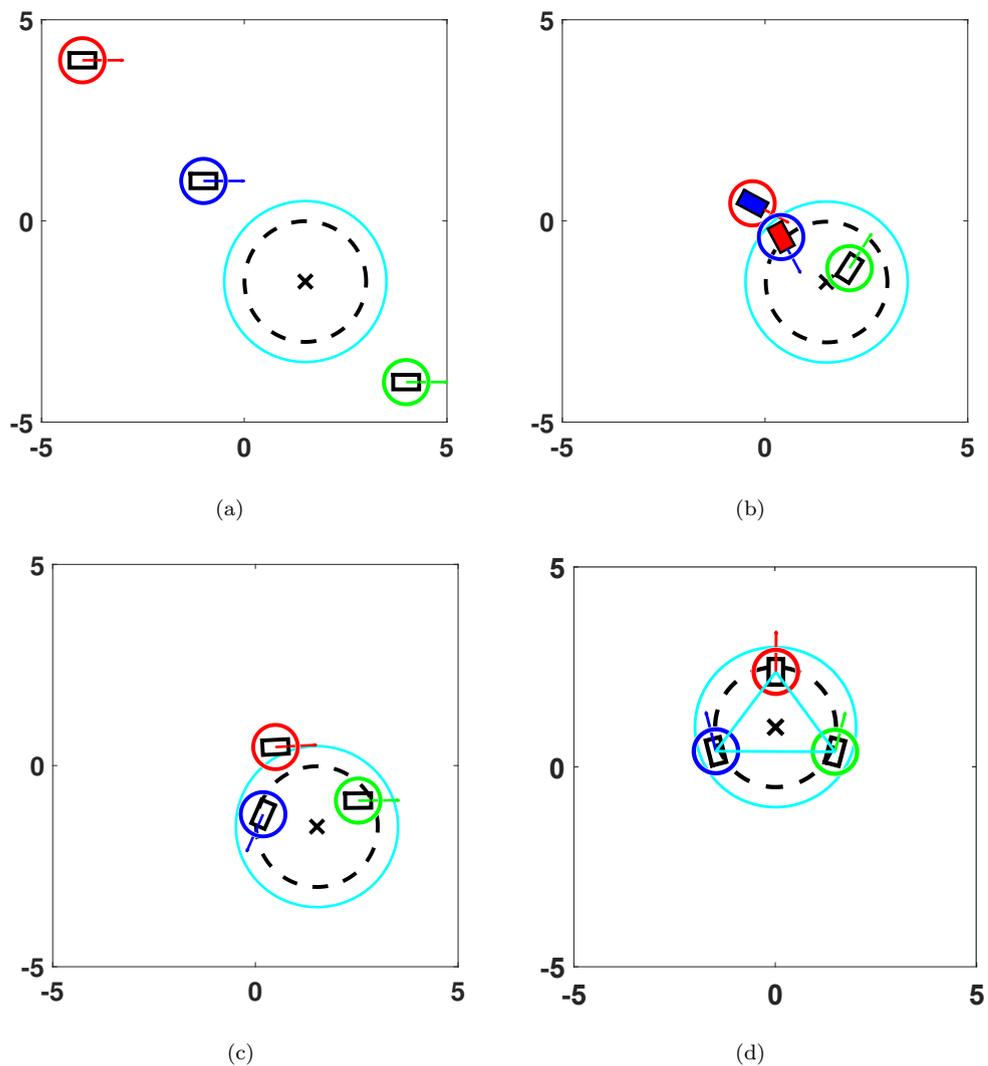
FIGURE 6.5: Scenario-3E: (a) initial config., (b) h-to-t collision detection, (c) collision avoidance (EC model) and (d) uniform formation

Robot R_1, in red, accelerates due to forces from other members of the group and goal attraction. Thus, it gains a high velocity that is then damped but not before a collision scenario emerge. With elastic collision model Fig. 6.5, the hitting robot slows down while pushing the robot in front along their radial axis. Soon after the collision is escaped, the robots continue following their planned trajectory and surround the goal.

Despite that the formation was constructed and the task was achieved in both scenarios, the LBR model exhibits an inefficient behavior of turning the colliding robots backwards as shown in Fig. 6.6.
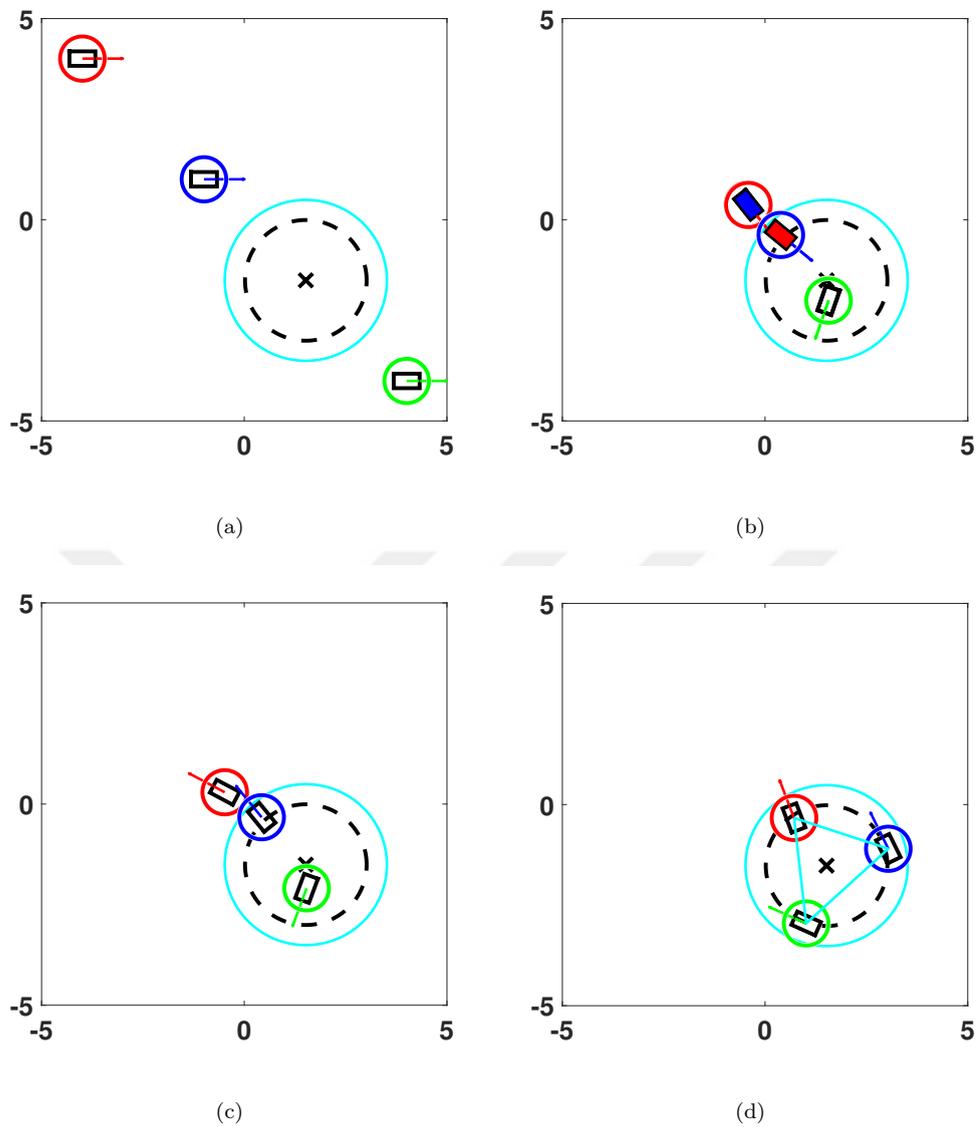
FIGURE 6.6: Scenario-3R: (a) initial configuration, (b) H-to-T collision detection, (c) collision avoidance (LBR model) and (d) uniform formation

A perception-based solution can be proposed to recognize the type of collision (h-to-h or h-to-t) and accordingly preventing the robot who've been hit from behind from activating its response. However, such solutions will negatively affect its major advantage of lower communication needs.

Before moving to four, five and ten robots' simulations, controllers performance is presented by the means of desired vs. controlled state curves for each of the aforementioned scenario [1-3]:
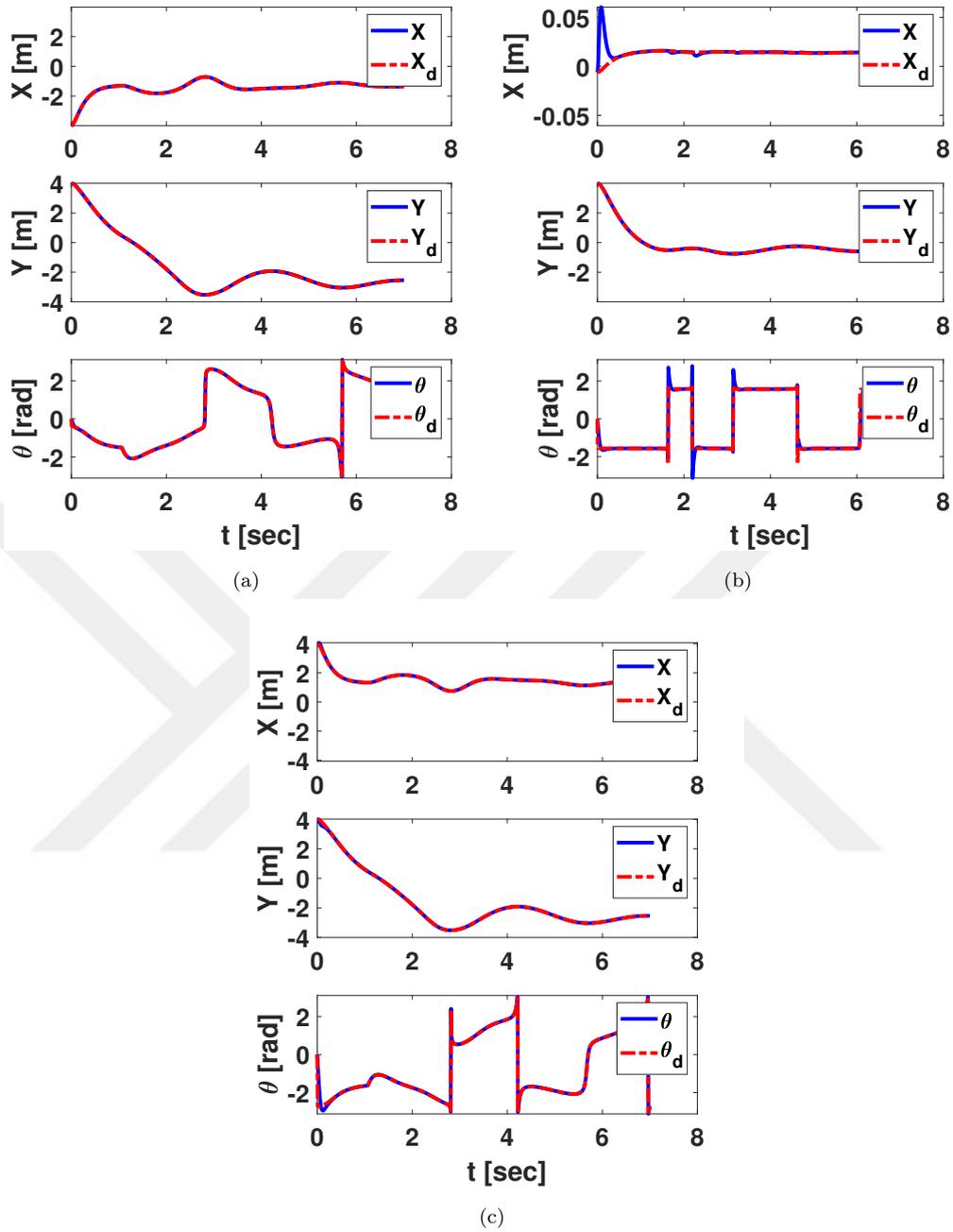
(a)

(b)

(c)

FIGURE 6.7: Scenario-1: position and orientation tracking performance (a) R_1, (b) R_2 and (c) R_3

Error	R1	R2	R3
e_x (m)	0.0017	0.0080	0.0174
e_y (m)	0.0038	0.0009	0.0285
e_θ (rad)	0.0261	0.2403	0.1641

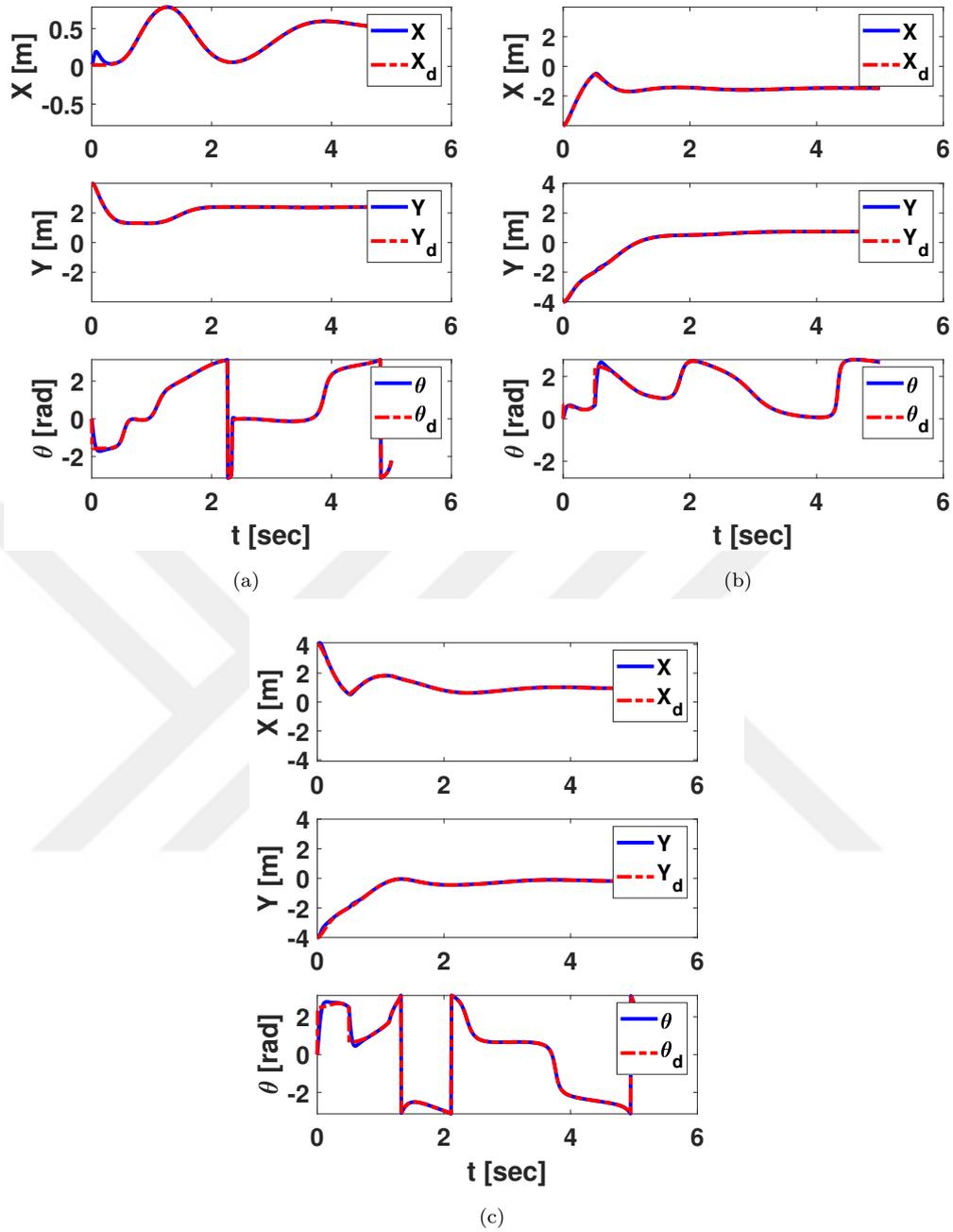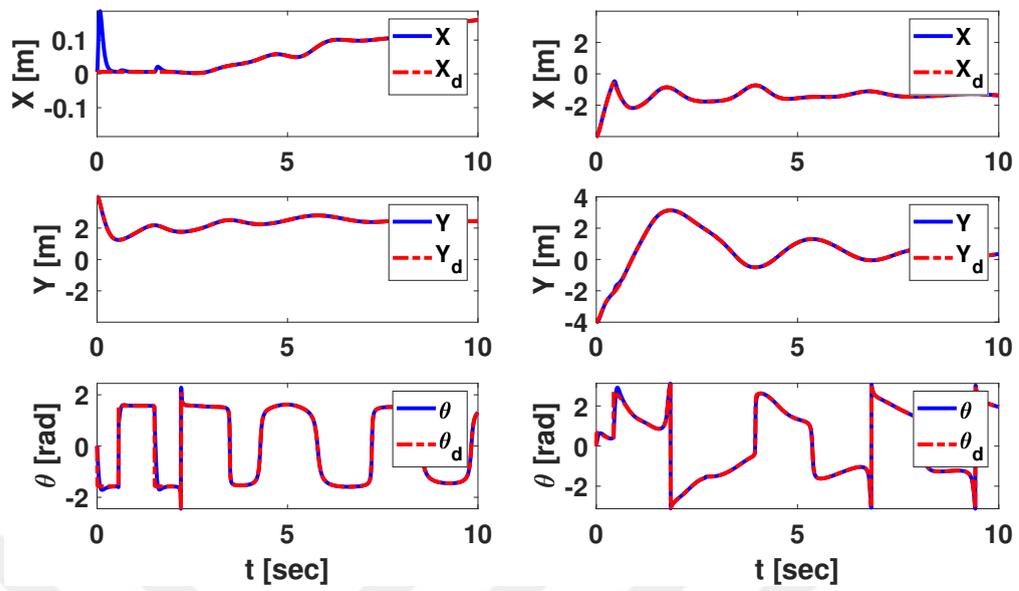TABLE 6.3: Scenario-1: RMS errors with virtual inputs controllers

(a)

(b)

(c)

FIGURE 6.8: Scenario-2E: position and orientation tracking performance (a) R_1, (b) R_2 and (c) R_3

Error	R1	R2	R3
e_x (m)	0.0244	0.0147	0.0347
e_y (m)	0.0034	0.0168	0.0398
e_θ (rad)	0.1171	0.1376	0.2183

TABLE 6.4: Scenario-2E: RMS errors with virtual inputs controllers

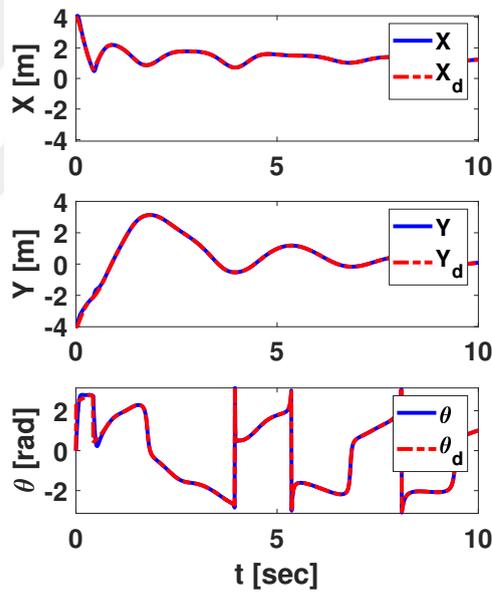FIGURE 6.9: Scenario-2R: position and orientation tracking performance (a) R_1, (b) R_2 and (c) R_3

Error	R1	R2	R3
e_x (m)	0.0177	0.0183	0.0291
e_y (m)	0.0025	0.0251	0.0363
e_θ (rad)	0.1712	0.1258	0.1733

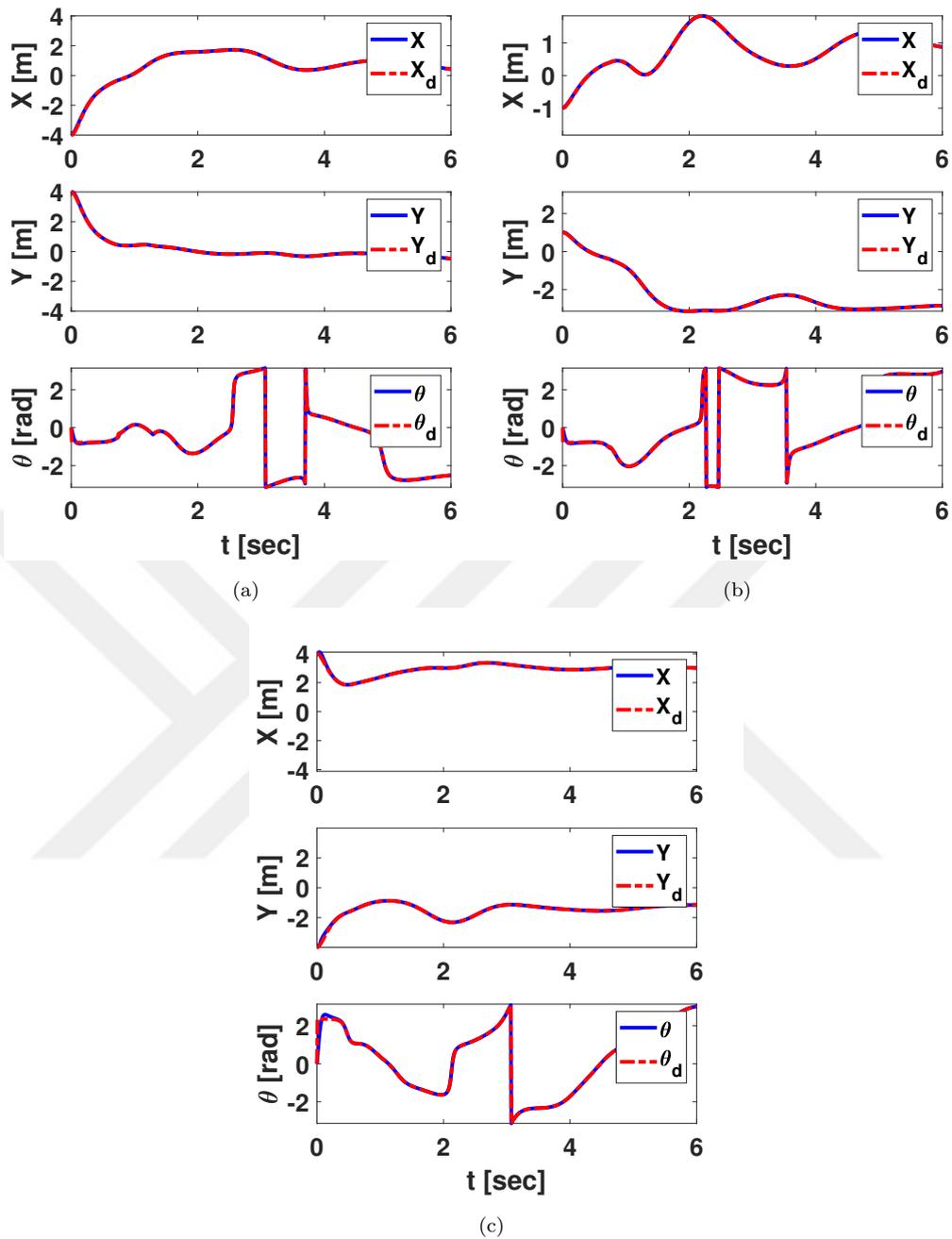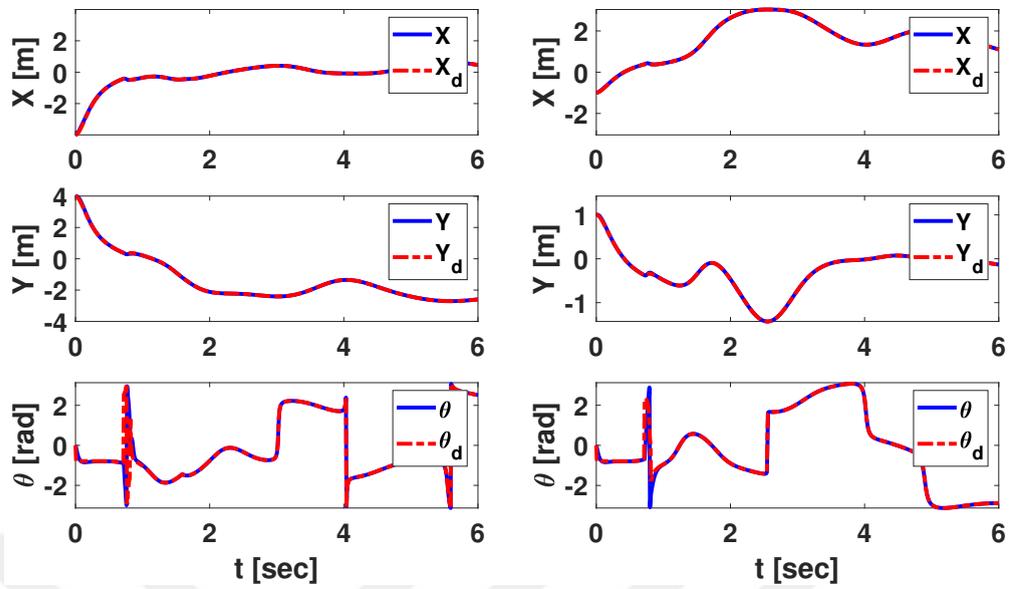TABLE 6.5: Scenario-2R: RMS errors with virtual inputs controllers

FIGURE 6.10: Scenario-3E: position and orientation tracking performance (a) R_1, (b) R_2 and (c) R_3

Error	R1	R2	R3
e_x (m)	0.0103	0.0030	0.0321
e_y (m)	0.0025	0.0030	0.0287
e_θ (rad)	0.1712	0.0537	0.1541

TABLE 6.6: Scenario-3E: RMS errors with virtual inputs controllers

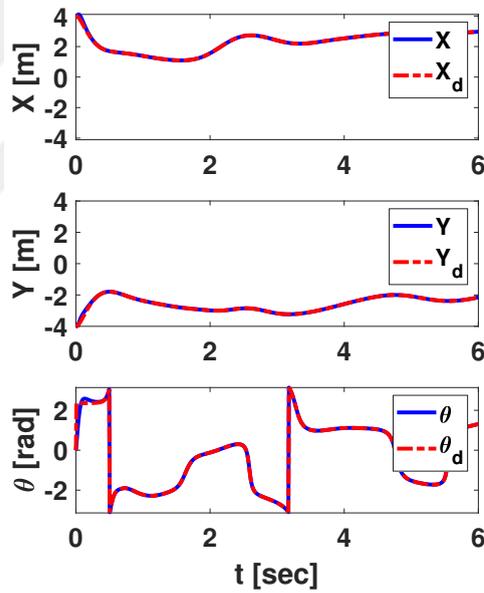FIGURE 6.11: Scenario-3R: position and orientation tracking performance (a) R_1, (b) R_2 and (c) R_3

Error	R1	R2	R3
e_x (m)	0.0107	0.0072	0.0322
e_y (m)	0.0150	0.0042	0.0288
e_θ (rad)	0.2157	0.2188	0.1539

TABLE 6.7: Scenario-3R: RMS errors with virtual inputs controllers

Scenario 4

In this scenario, four robots are aligned horizontally on the top while the goal is on the bottom of the workspace. From now on, we utilize EC model and only show frames from motion's animation. System's parameters are the same as scenario 3 except for the initial robot positions and goal position.

Two instances of collisions happened between robots as depicted in Fig. 6.12. First, a collision is detected and avoided between blue and green robots' shells, Then, red and magenta colored robots also undergo a *safe collision* of their shells. Despite successfully working, modular structure of the system was not fully clear due to low number of robots.

Scenario 5

Five robots are used for further investigating scalability property of the system. As the number of the robots increased, we relax goal conditions in terms of d_G circle radius around the goal, so that the robots can fit in. System's modified parameters can be found in Tab. 6.8. The robots are distributed along the edges of the 2D environment, while the goal is seated in the southeast corner of it.

Parameter	Value
d_G	2
d_{break}	3
d_R	2
k_R	20
c_R	4

TABLE 6.8: Scenario-5: modified parameters

Triangular mesh formed by the two nearest neighbors approach is very clear in this scenario, Fig. 6.13. This result is promising for large scale swarms of robots. Another important notice is that this scenario would have ended with four defected robots out of 5, if it was not for the collision avoidance framework. In this case of multiple-collisions, framework choose the response with the highest magnitude, thus responding to first to the more possibly crucial collision.
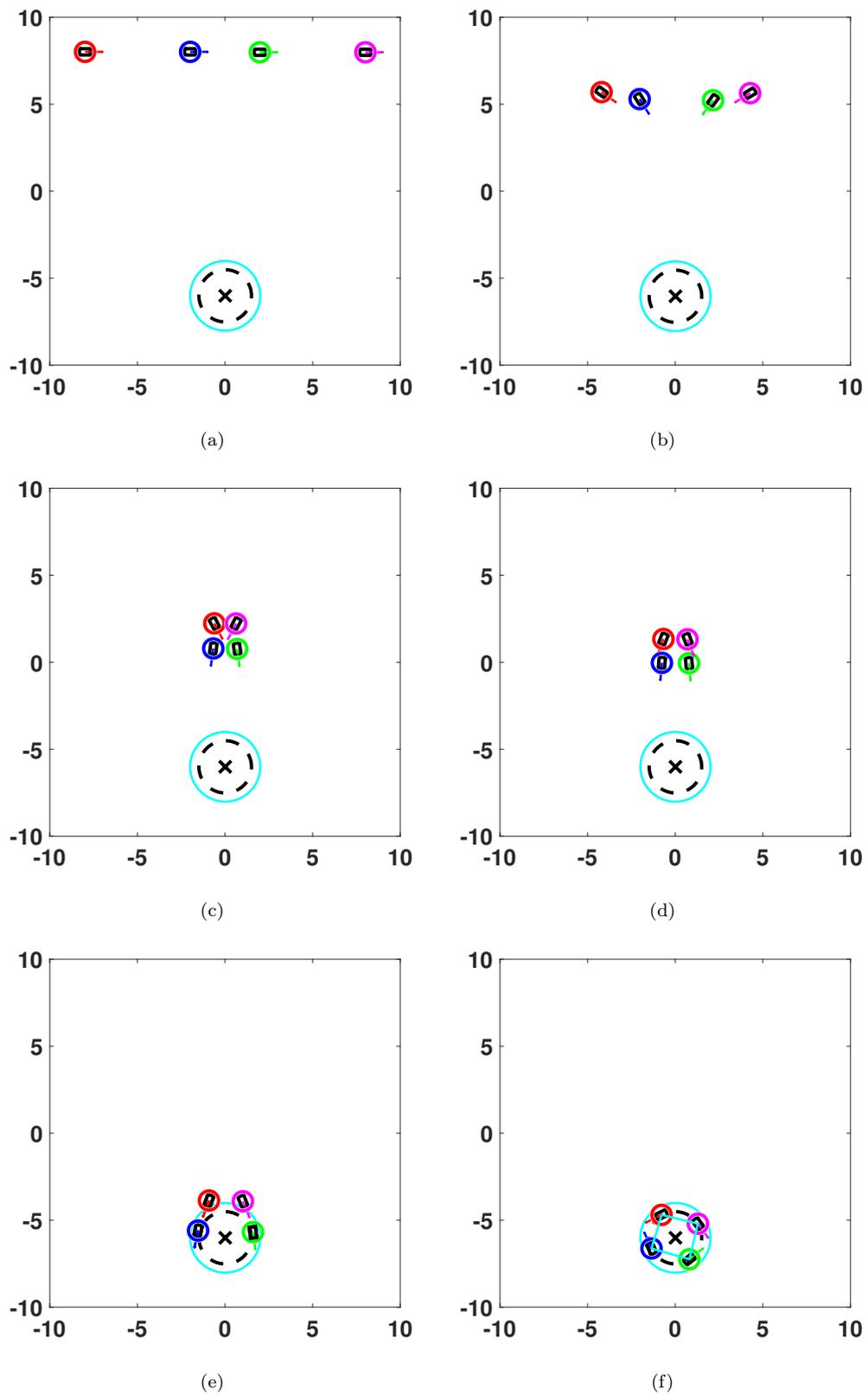
FIGURE 6.12: Scenario-4: (a) initial config., (b) coordination dominance, (c) 1st collision avoidance, (d) 2nd collision avoidance, (e) virtual bonds relaxation, and (d) uniform square formation
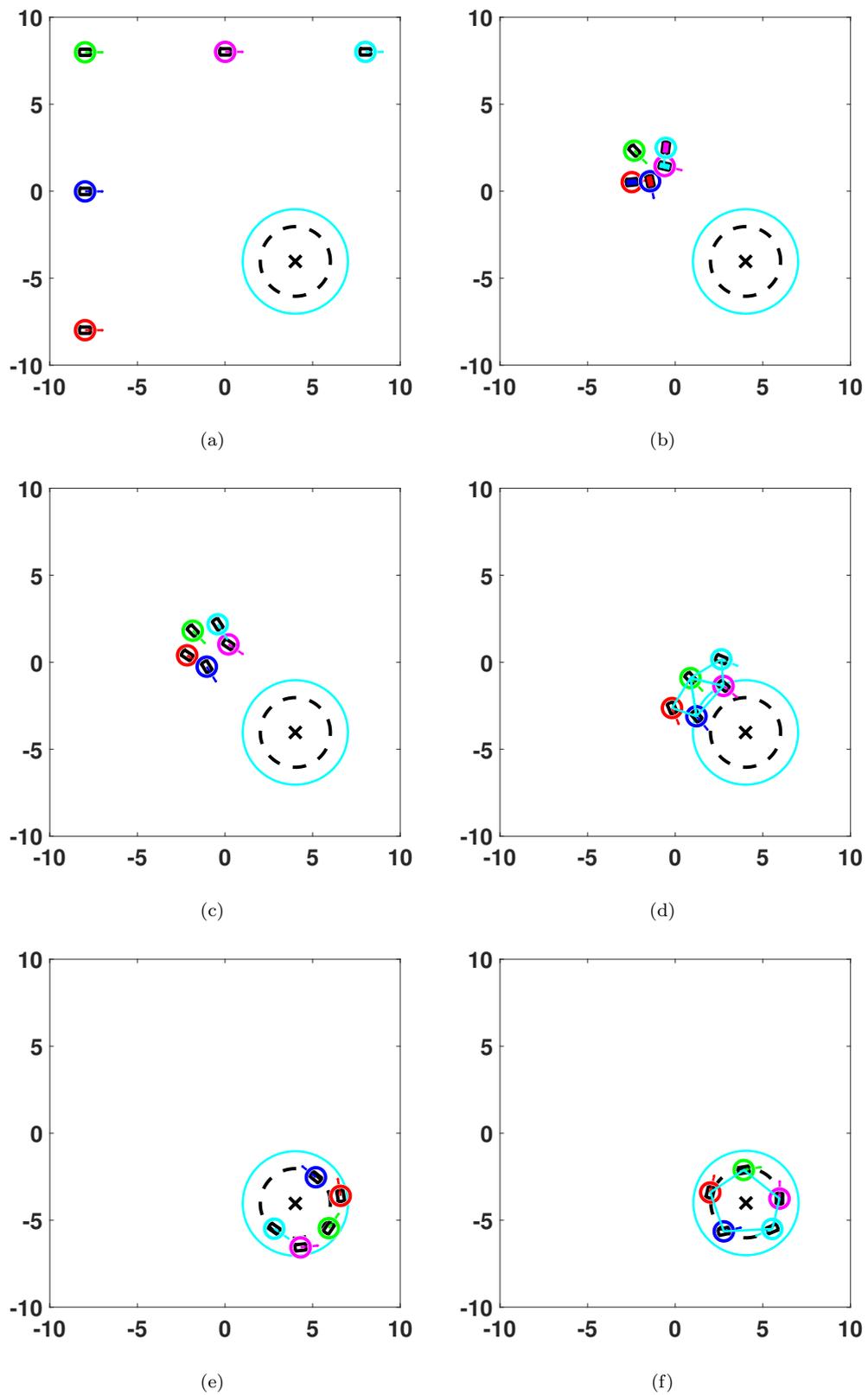
FIGURE 6.13: Scenario-5: (a) initial config., (b) multiple-collision detection, (c) all collision avoided, (d) G approaching while forming triangular mesh formation, (e) relaxing bonds (d) uniform pentagon formation

Scenario 6

Ten robots are used to mimic realistic swarm scenarios and test system's performance accordingly. Robots are uniformly distributed on a line to the north of the workspace, with the goal at the most south of it. This scenario is analogical to real-life swarm tasks where generally robot group initially start from near base stations and head towards the goal. Few system parameters where modified to adapt with the large number, Tab. 6.9.

Parameter	Value
d_G	3
d_{break}	4
c_R	4

TABLE 6.9: Scenario-6: modified parameters

Due to large number of successive collisions per robot, we use a bar graph to interpret collision data. A total of 36 collisions was detected and avoided with robots in the middle being exposed to the largest number of collisions Fig. 6.17. The trajectory profile in Fig. 6.14 indicates how the coordinated motion switches between the approaching phase, where coordination forces are dominant, and formation construction phase, where goal convergence and formation distance maintenance forces are dominant.

Scenario 7

As a final bench scenario, ten robots in a platoon formation along the main diagonal of the workspace are intended to surround a goal in its sight. Initial configuration is shown in Fig. 6.14

A worm motion-similar successive expansions and shrinkage occur in coordination phase. Large number of collisions are avoided Fig. 6.17, thus enabling the robots to start form their triangular module formations with their nearest neighbors. Generated trajectories shows a smooth transition from the platooning state to the new coordination state. Thus, this task can be stitched to other tasks to perform an overall mission e.g. moving from one neighbor to another in platoons and then surrounding a fire to extinguish it.
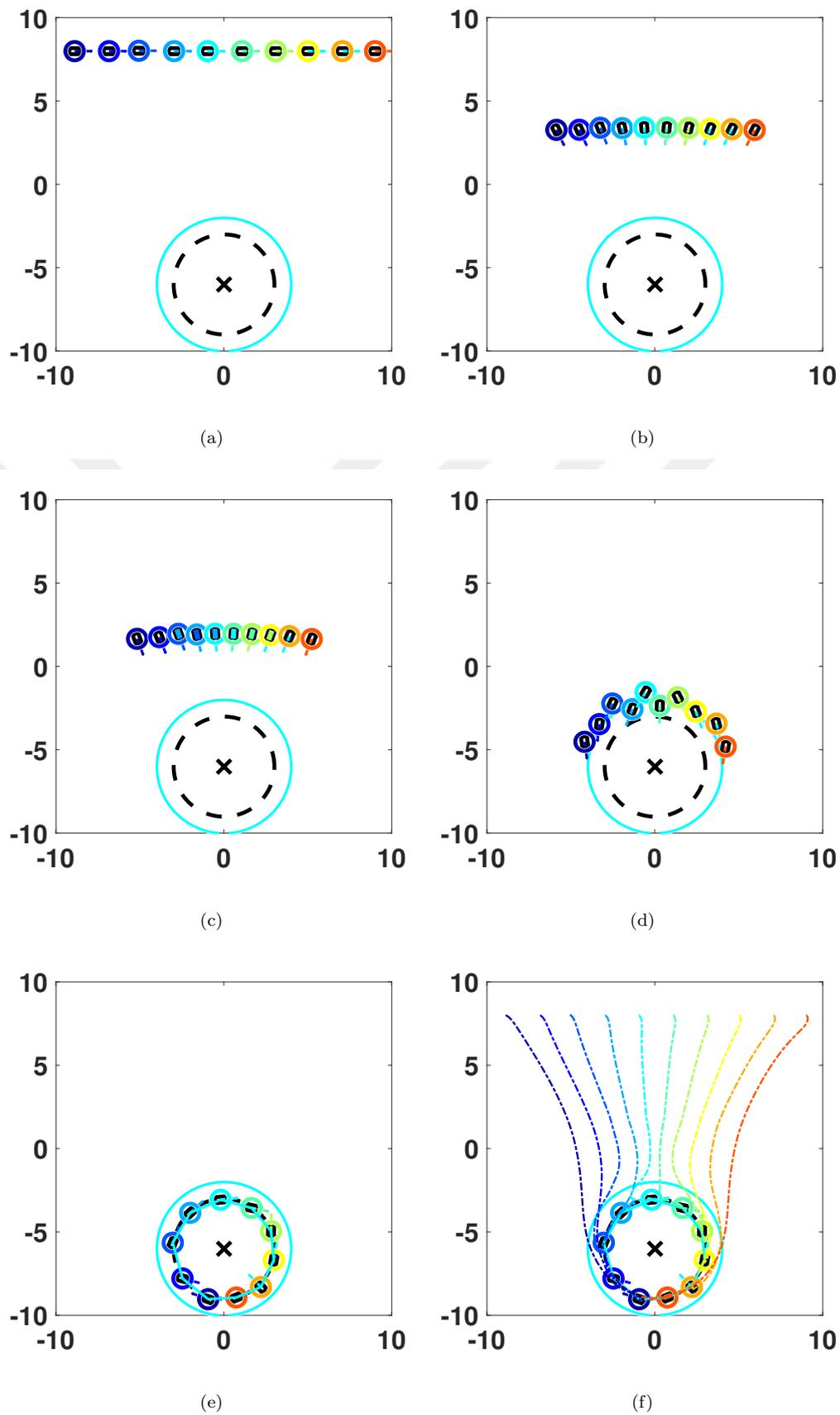
FIGURE 6.14: Scenario-6: (a) initial config., (b) coordinated motion, (c) multiple-collisions detected, (d) collisions avoided, (e) decagon uniform formation achieved and (d) robots' trajectories
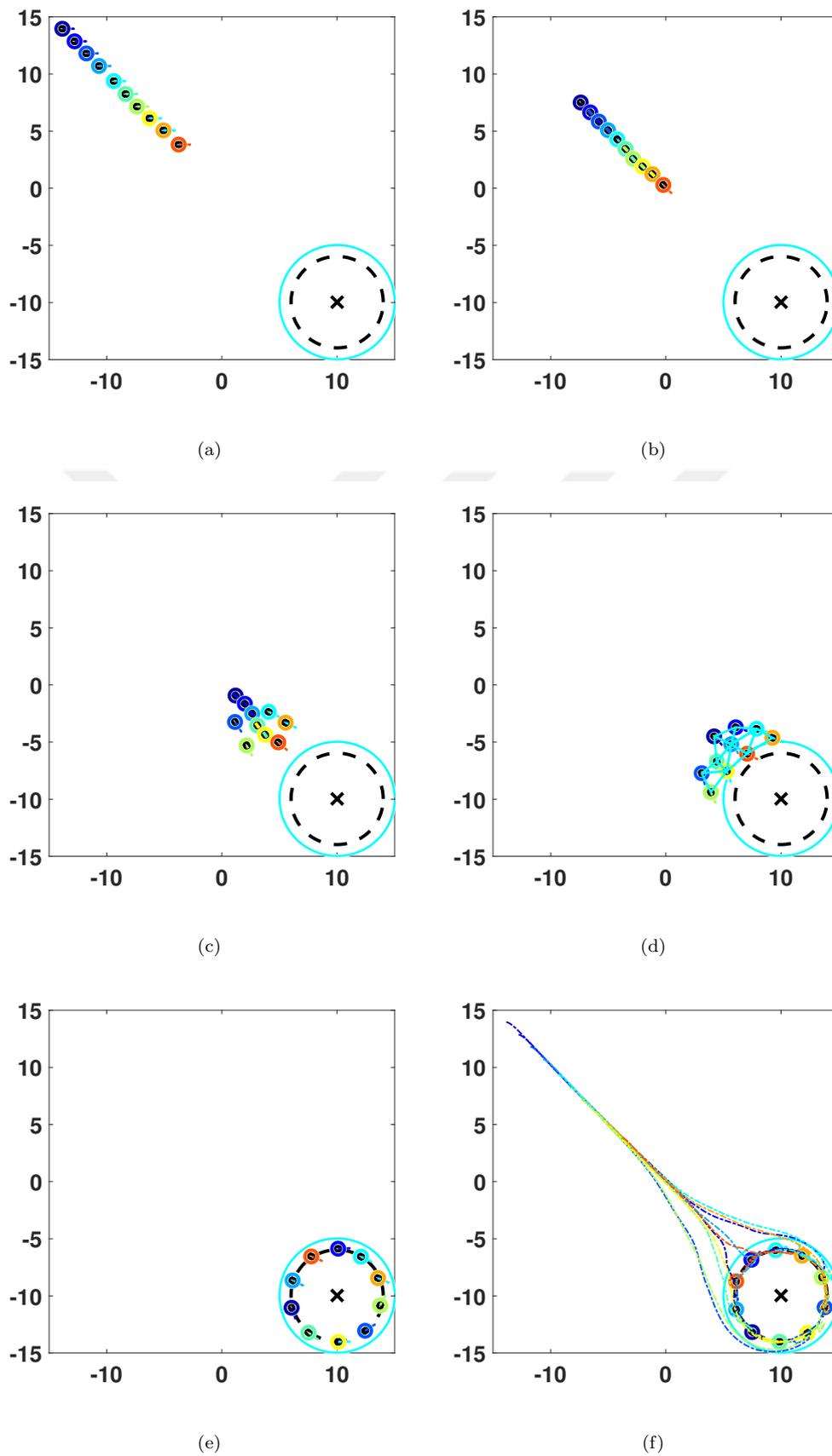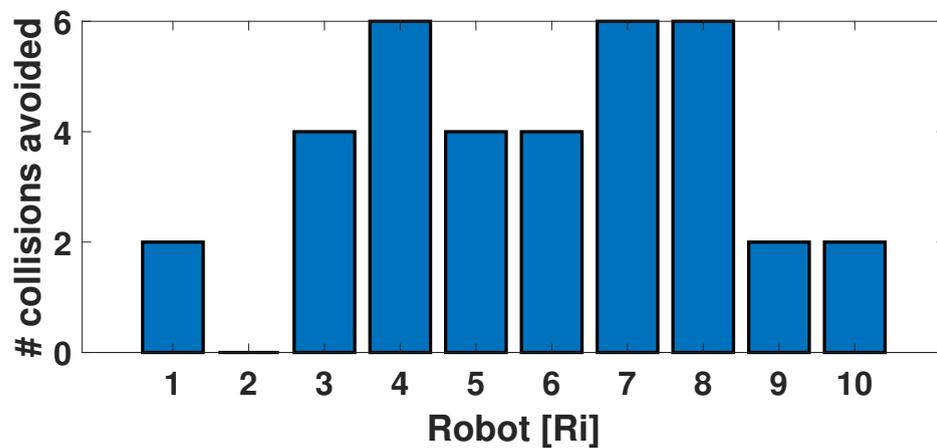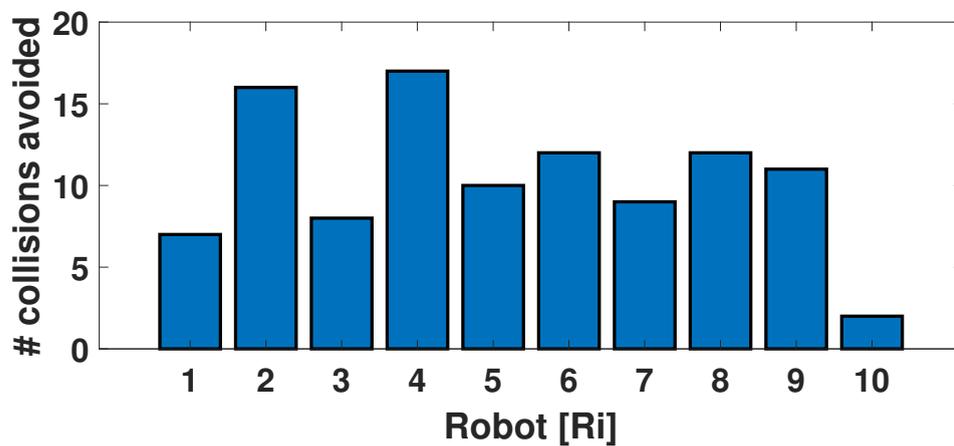
FIGURE 6.15: Scenario-7: (a) initial config. (b) coordination forces dominant (c) 1st collision avoidance (d) 2nd collision avoidance (e) relaxing bonds (d) uniform square formation

FIGURE 6.16: Scenario-6: R_i vs. number of avoided collisions

FIGURE 6.17: Scenario-7: R_i vs. number of avoided collisions

The results of the simulations for the proposed collision avoidance framework embedded in the general coordination structure with three, four, five and ten robots are all satisfactory as discussed earlier. Scalability, robustness and scenario-independence was positively verified.

6.2　3D Workspace implementation

The simulations were carried out for groups of three and five quadrotor Type UAVs. In a similar approach to the previous section, we first demonstrate the operation of system's components, then move to a more realistic and challenging scenarios. In the following simulations, virtual masses are all set to $m_i = 0.8 \, [kg]$, quadrotors' have identical dynamic parameter values, arm length of the quadrotor is $l = 0.3 \, [m]$ and accordingly virtual shells $\Omega_i(r_i)$s' radii were set to $r_i = 1.2 \, [m]$ with a high safety factor β considering its high operating speeds. However, size of the quadrotor in the following snapshots of the simulations does not reflect its real size, we enlarge it to be one sixth of the cubic workspace side length for flight characterization visualization purposes. Controller values, obtained experimentally, are the same for the controllers of all quadrotors Tab. 6.10. Moreover, virtual reference trajectory generation systems parameters are set in Tab. 6.11, any scenario-based modifications for parameters will be explicitly stated.

Gains	X	Y	Z	ϕ	θ	ψ
K_p	3.2	4.1	4	3.1	7	3
K_i	0.001	0.001	0.001	0	0	0
K_d	5	5	7	5.5	15	7

TABLE 6.10: Quadrotor position and attitude controller gains

Parameter	Value
b	> 10
k_R	9
c_R	3
k_G	15
c_G	4
$k_R{}'$	2
α	10
γ	1

TABLE 6.11: Simulation parameters

Scenario 1

We first consider a collision-free scenario of a group of three quadrotors surrounding the goal at relatively close distances $Q_1 = [5\ 0\ 0]$, $Q_2 = [-5\ 5\ 0]$ and $Q_3 = [-5\ -5\ 0]$. The goal is fixed above the robots 5 meters on top of workspace origin, $G = [0\ 0\ 5]$. d_G is set to 3, $d_R = 2.5$ and $d_{break} = 4$. Planner's produced trajectories together with the controller actual ones are shown in Fig. 6.31.
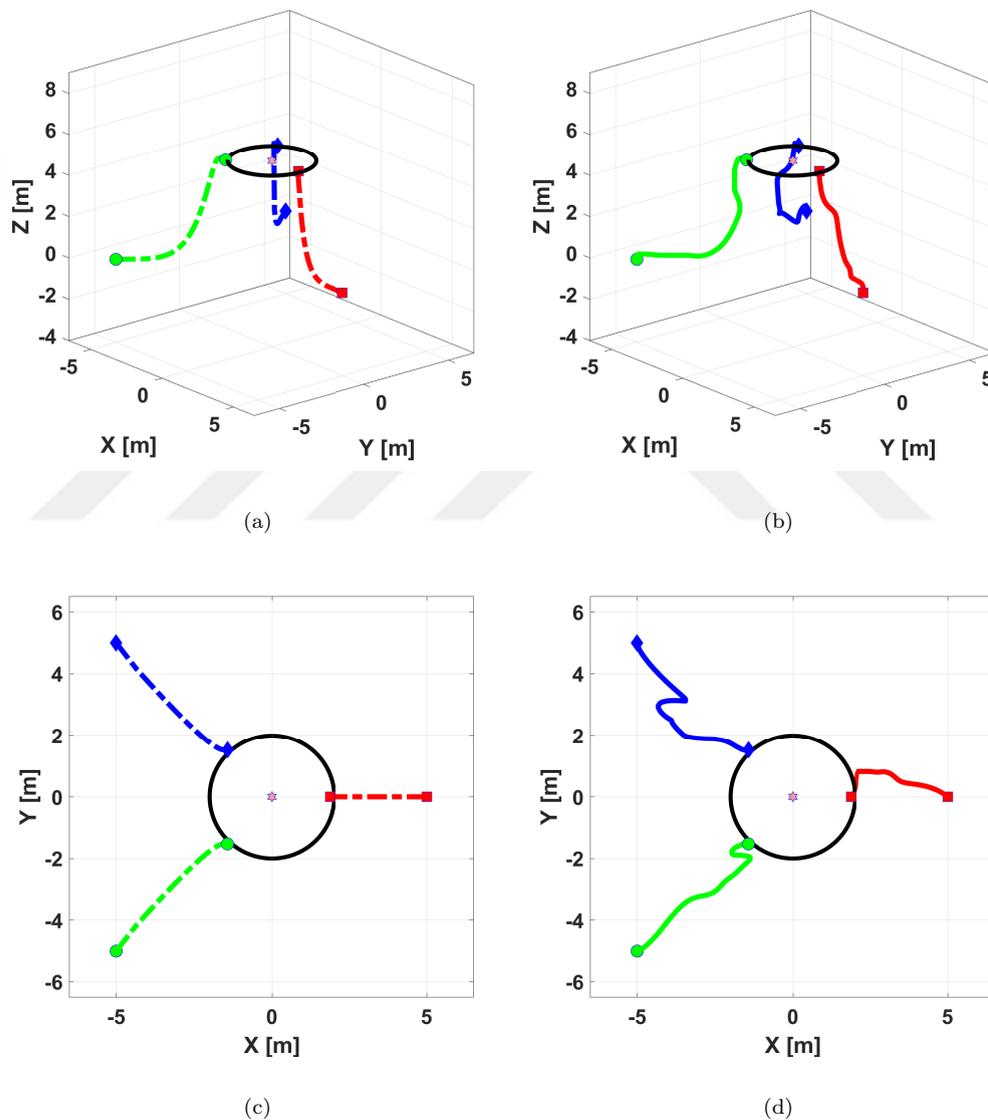

FIGURE 6.18: 3D scenario-1: (a) reference trajectories, (b) actual trajectories, (c) reference trajectories' projection on xy plane, (d) actual trajectories' projection on xy plane

Initial configuration in space is as illustrated in the animation snapshots of Fig. 6.32. The quadrotors take off and try to reach each other coordination forces dominance, while elevating towards G's altitude. Once entered d_{break} circle, bonds relaxes and the final formation is achieved.
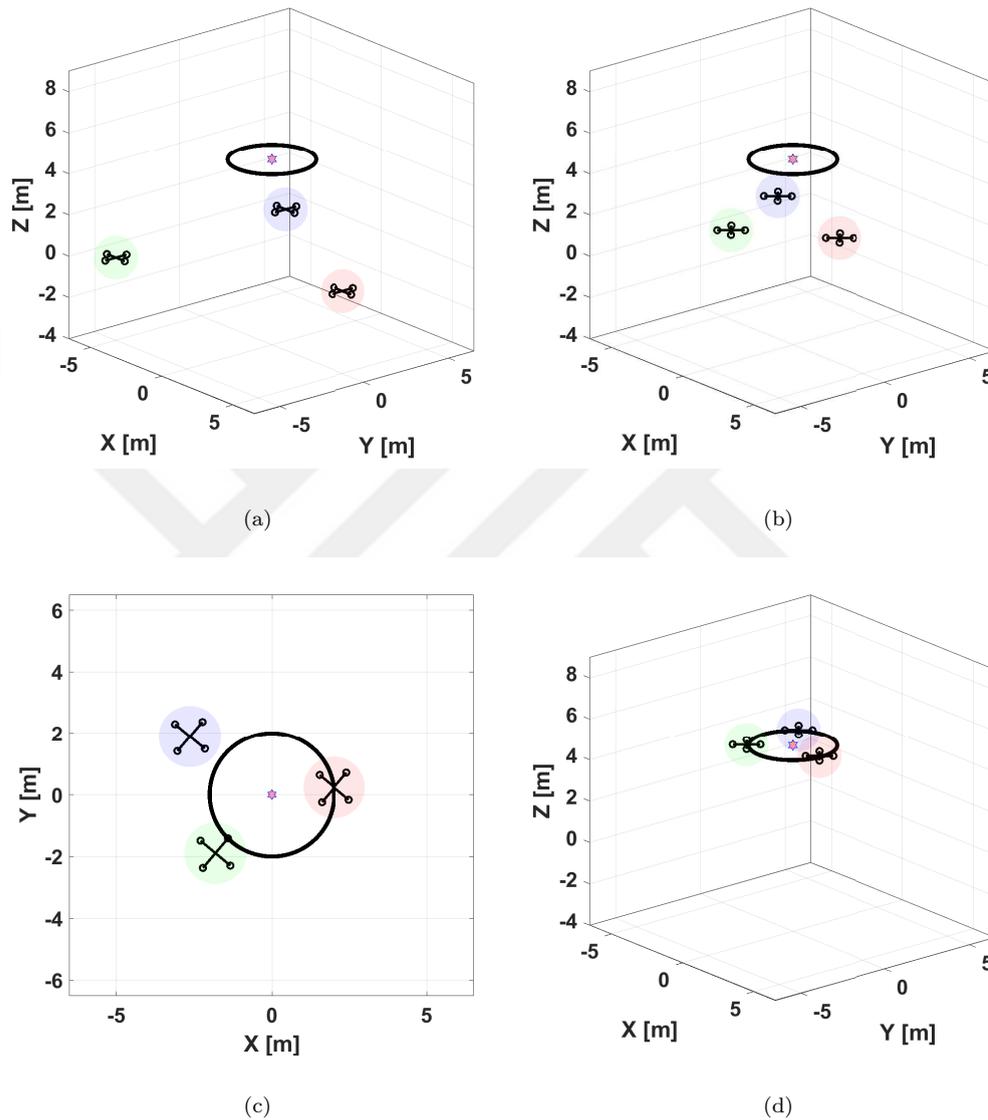

(a)

(b)

(c)

(d)

FIGURE 6.19: 3D scenario-1 (a) initial config., (b) approaching G in coordination, (c) spreading pre-formation , (d) final uniform formation around

Tracking performance is visualized in 6.20. An important notice is the perturbations in tracking planner's trajectory. One major reason is the smoothness of the reference trajectories. Despite having the Z component of the trajectory as a quintic polynomial smooth, the xy horizontal plane trajectories are not guaranteed to be so. This problem is bypassed by suitable planner parameters adjustment.
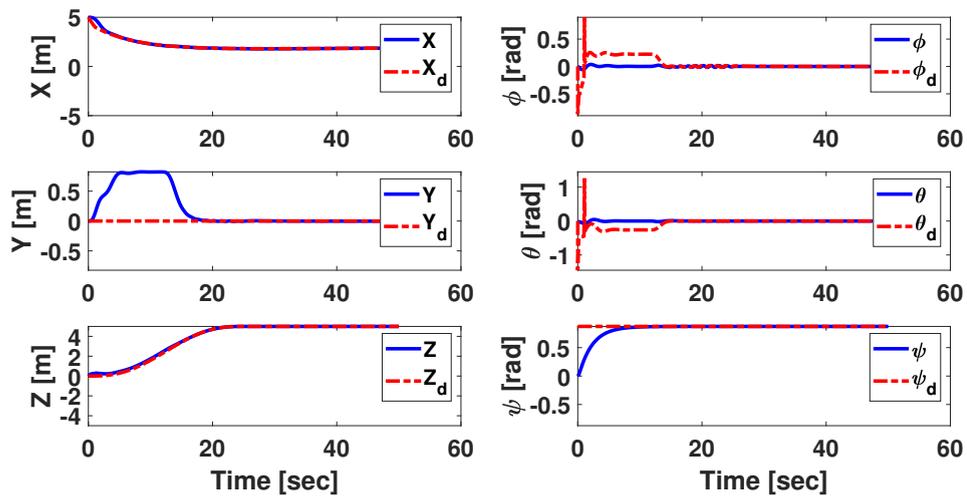
(a)
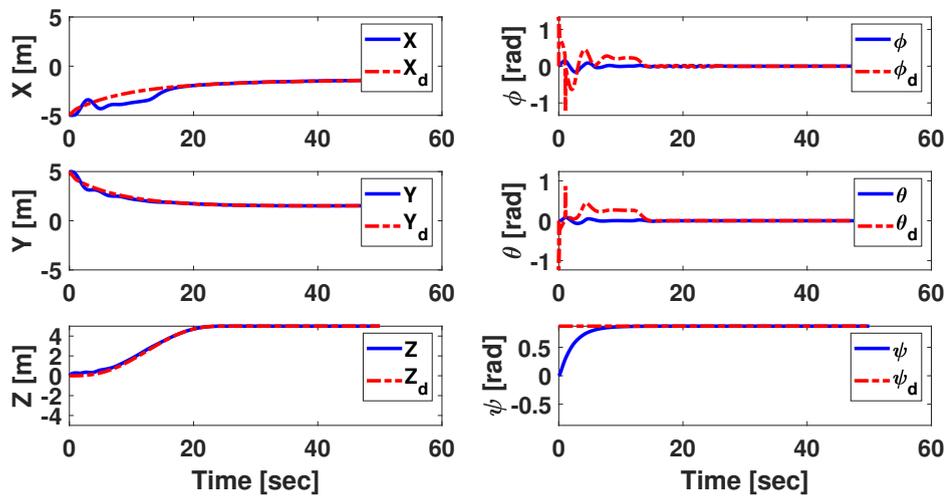
(b)
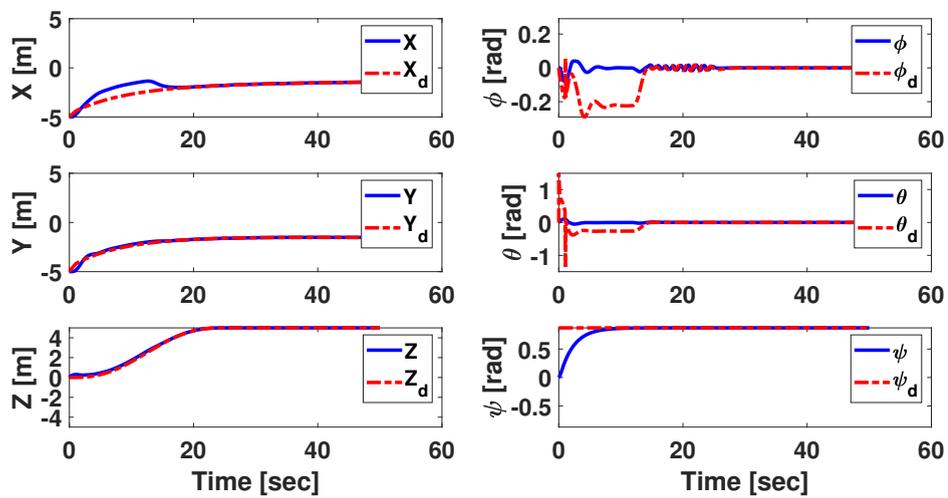
(c)

FIGURE 6.20: 3D scenario-1: position and orientation tracking performance (a) Q_1, (b) Q_2 and (c) Q_3

Scenario 2

We test collision avoidance framework by introducing a scenario where two quadrotors were lunched by relatively far station to group with the third member who detected the goal. As for system's modified parameters, $Q_1 = [-9\ 4\ 0]$, $Q_2 = [-9\ -4\ 0]$, $Q_3 = [3\ 0\ 0]$, $d_G = 2$, $d_{break} = 3$, $c_R = 3$ and $k_G = 5$.
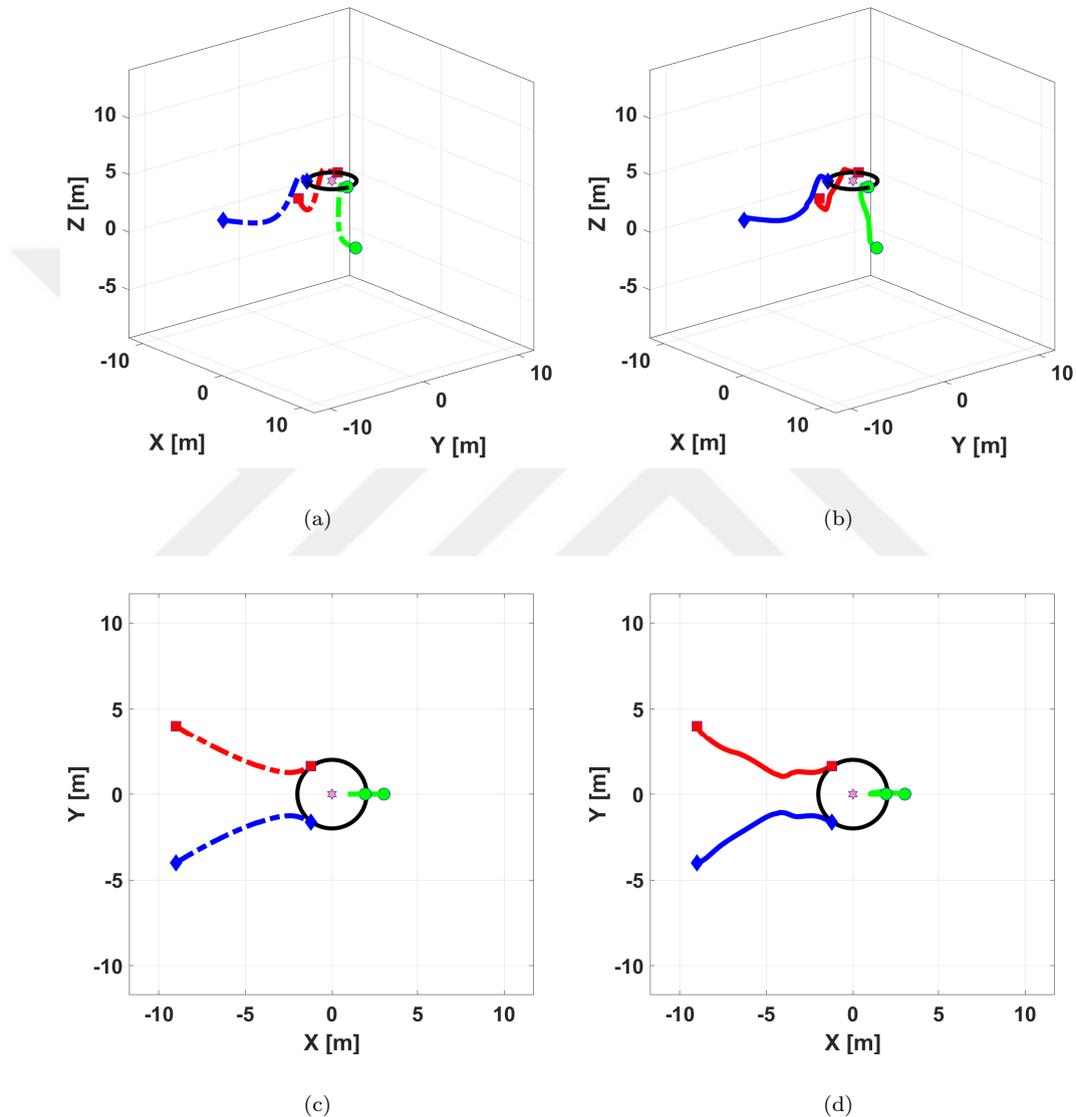

FIGURE 6.21: 3D scenario-2: (a) ref. traj., (b) actual traj., (c) ref. trajectories' projection on xy plane, (d) actual trajectories' projection on xy plane

A collision is detected and successfully avoided around $t = 10\ [sec]$ between the Q_1 and Q_2. Quadrotor control performance presented in Fig. 6.23 shows an error cusp at the collision occurrence indicating controller's effort to track the spontaneous change of desired trajectory to avoid the collision.
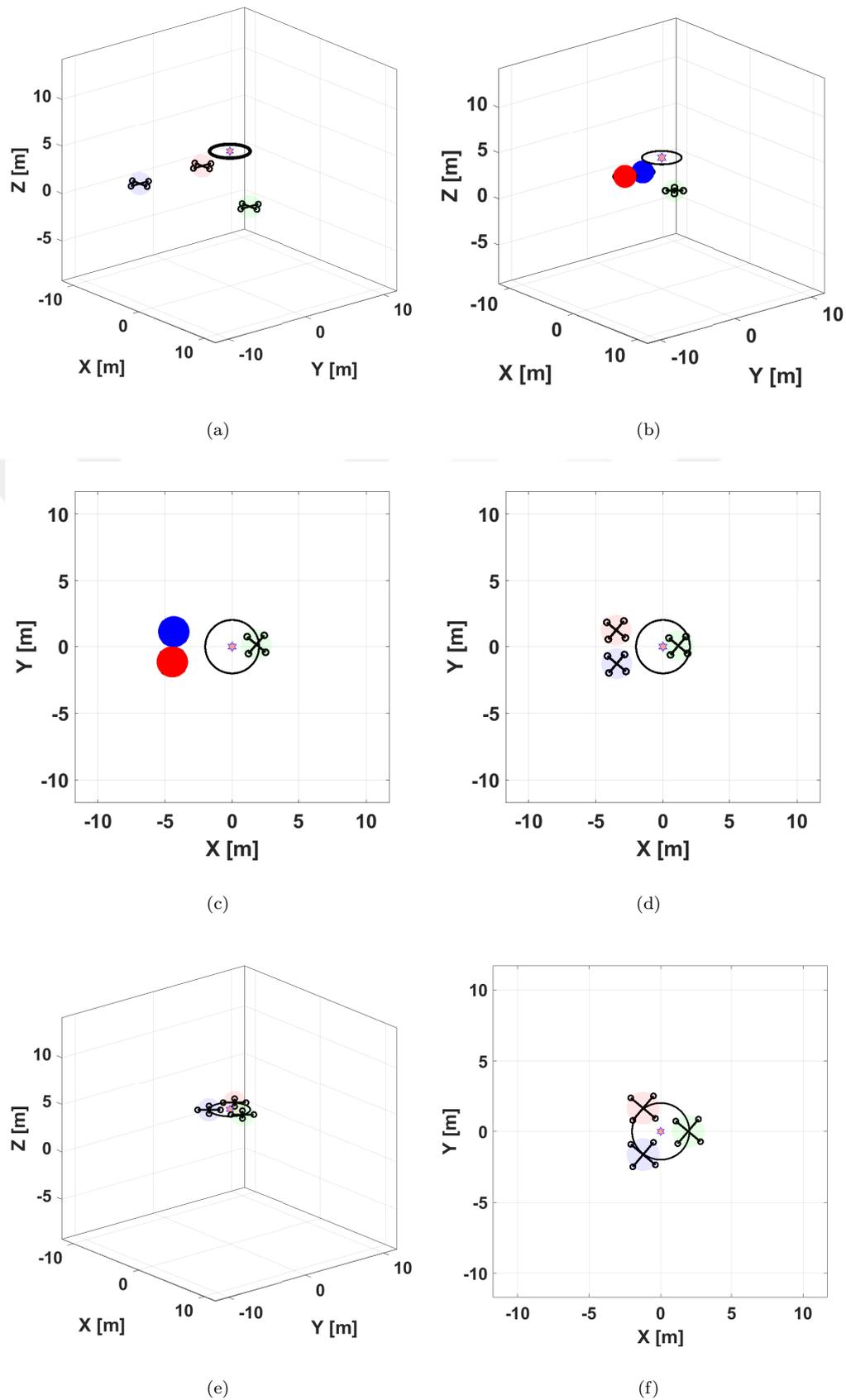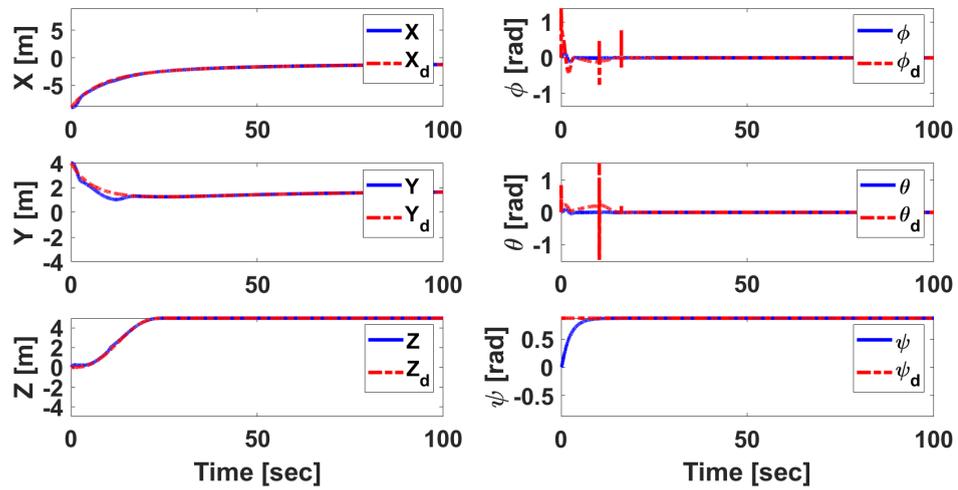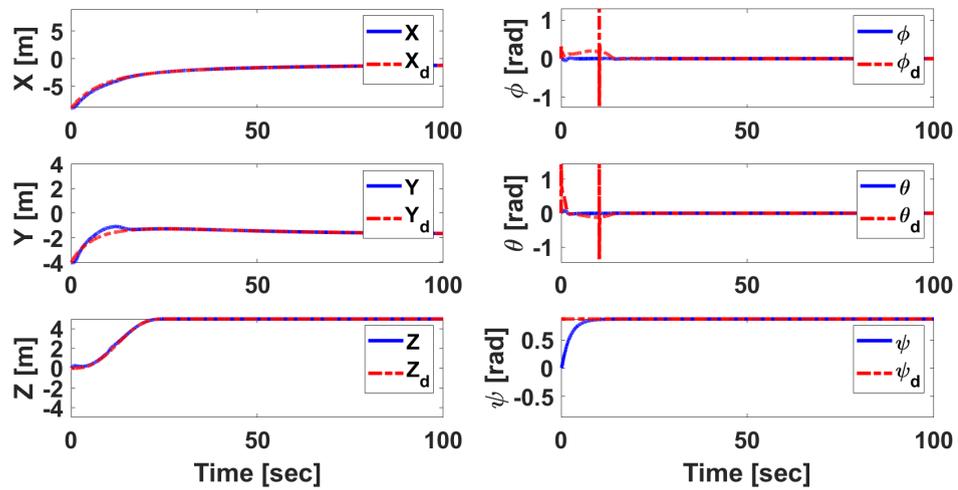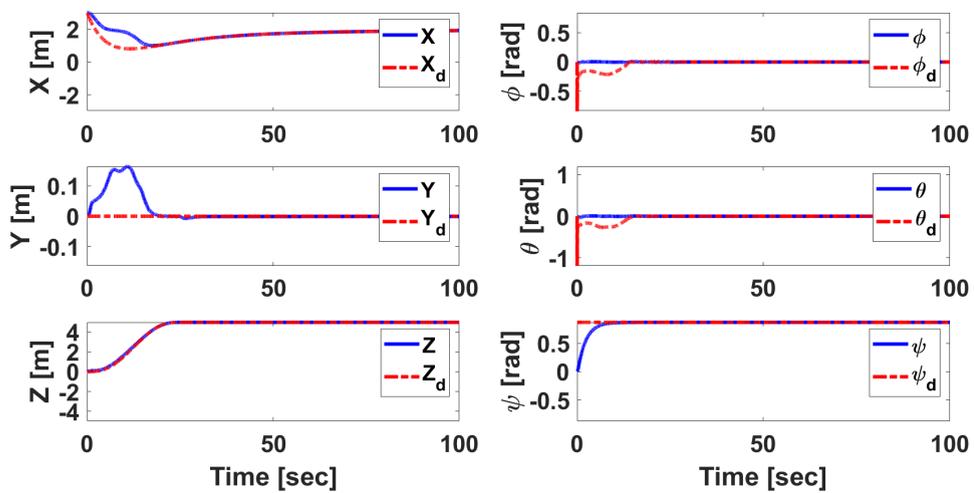
FIGURE 6.22: 3D scenario-2 (a) initial config., (b) collision detection: perspective view, (c) collision detection: top view, (d) collision avoided using EC model, (e) relaxing bonds after d_{break} and (f) achieving task formation

FIGURE 6.23: 3D scenario-2: position and orientation tracking performance (a) Q_1, (b) Q_2 and (c) Q_3

Scenario 3

A realistic scenario of three quadrotors trying to reach and explore a target from a distance of 10 $[m]$ is investigated. Similar system parameters as the previous scenario are being used.

(a)

(b)

(c)

(d)

FIGURE 6.24: 3D scenario-3: (a) ref. traj., (b) actual traj., (c) ref. trajectories' projection on xy plane, (d) actual trajectories' projection on xy plane

Multiple collisions through out the full trajectory is demonstrated in this scenario where two sequential collisions occur. Virtual mass model of the high-level planner was fed by the newly calculated post-collision velocity and an online change of desired path is achieved. Finally, controller sufficiently tracks the modified trajectory thus avoiding collisions. To conclude, simulations with three quadrotors exhibited satisfactory results with the condition that the trajectory is smooth enough.
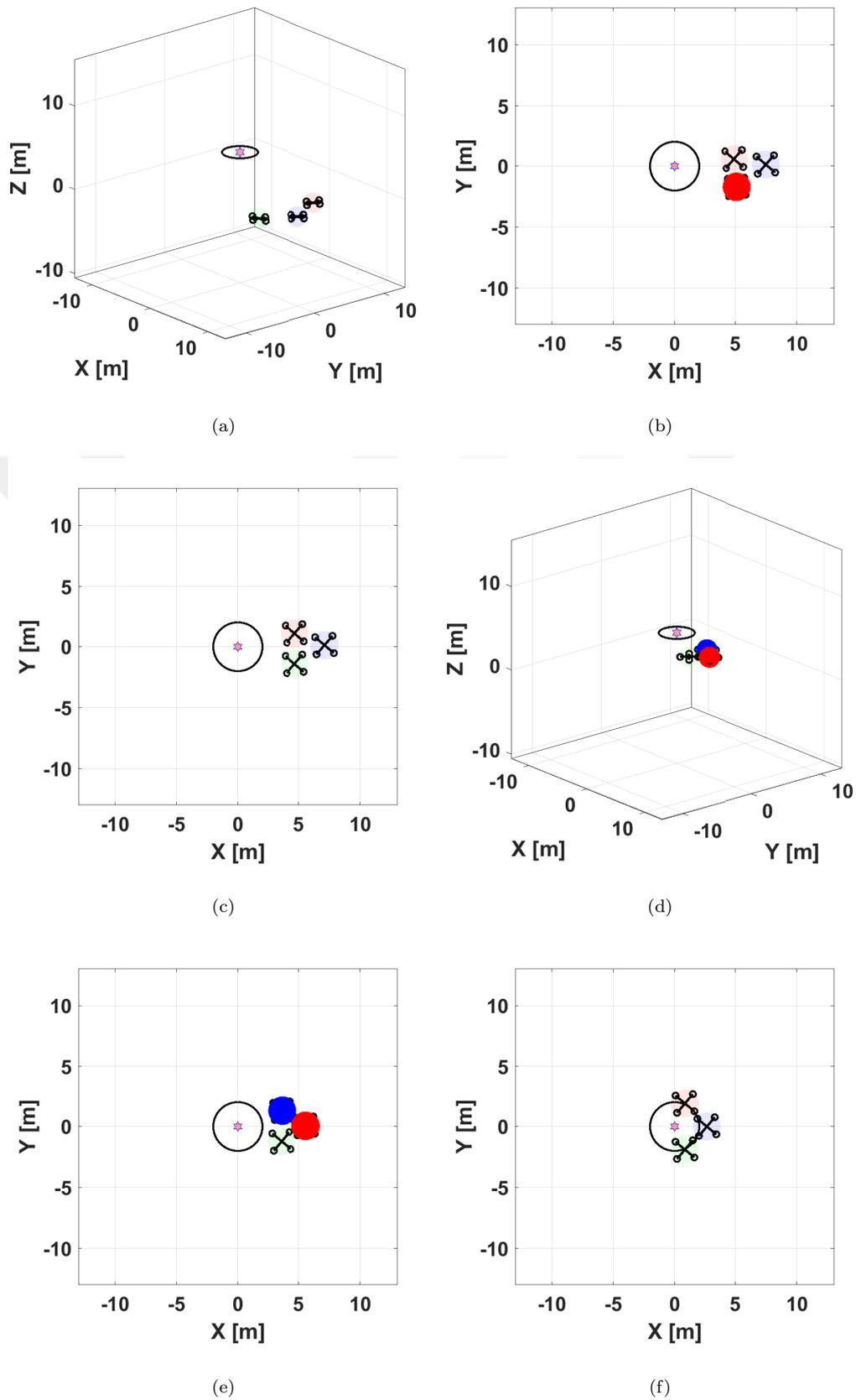
FIGURE 6.25: 3D scenario-3 (a) initial config., (b) 1st collision detection: top view, (c) collision avoided, (d) 2nd collision detected: perspective view, (e) 2nd collision detected: top view and (f) achieving task formation
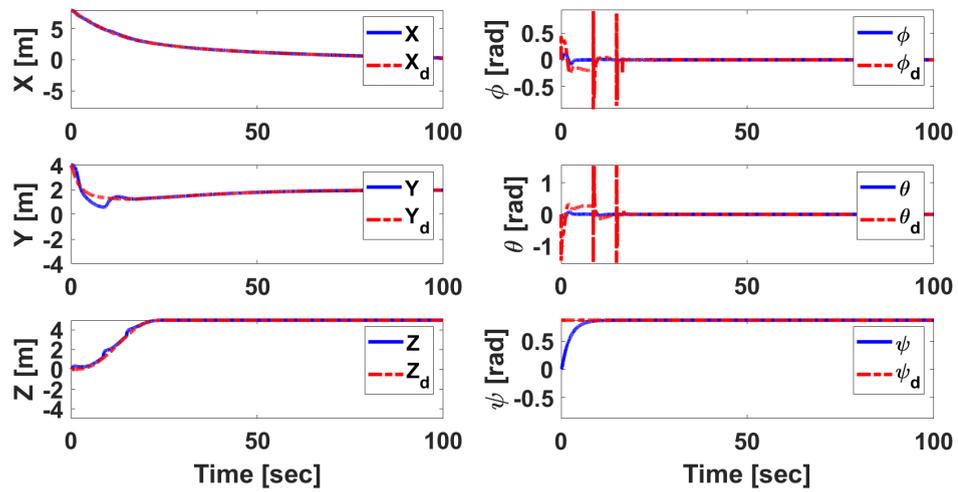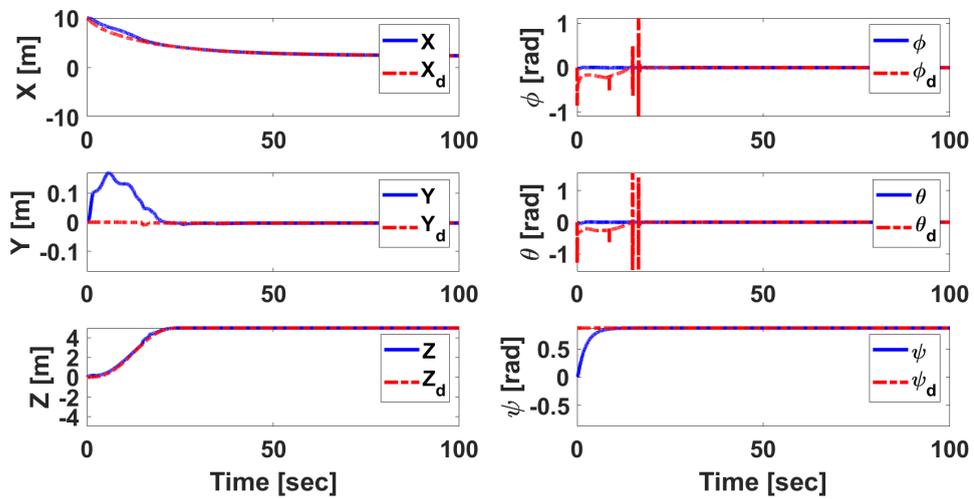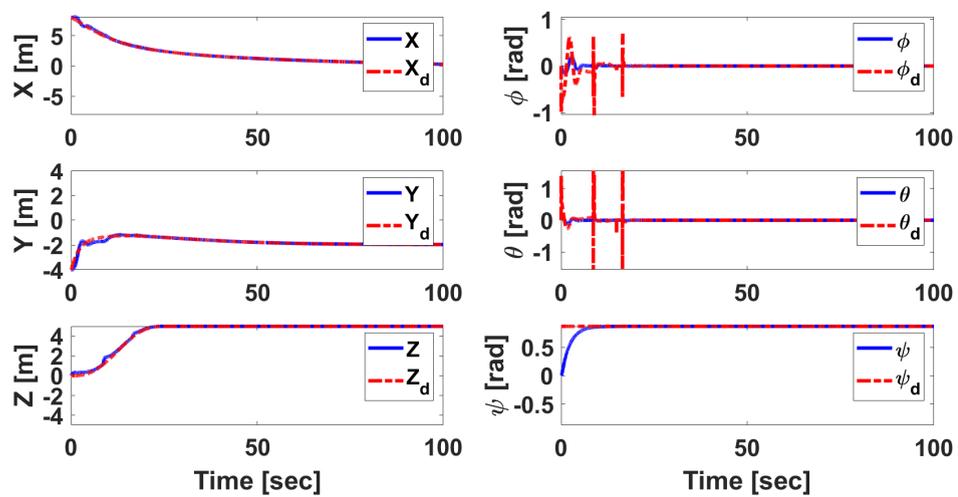
FIGURE 6.26: 3D scenario-3: position and orientation tracking performance (a) Q_1, (b) Q_2 and (c) Q_3

Scenario 4

We investigate system's overall performance and modular structure with five quadrotor UAVs group. This scenario mimics **scenario 1**, with the exception of increased collision potential due to increased number of robots. Planned and followed trajectories are depicted in Fig. 6.27 below:
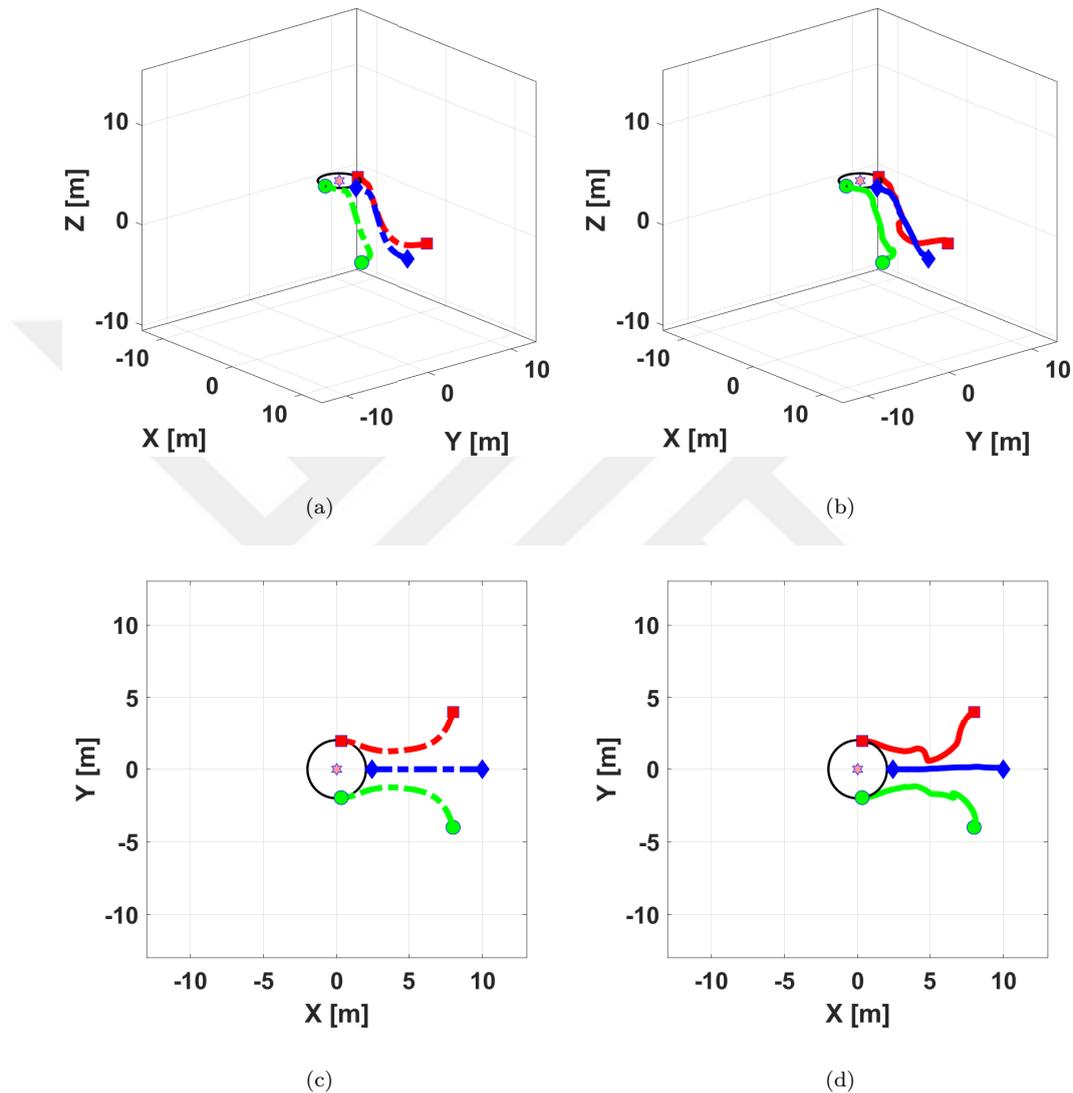

FIGURE 6.27: 3D scenario-4: (a) ref. traj., (b) actual traj., (c) ref. trajectories' projection on xy plane, (d) actual trajectories' projection on xy plane

Quadrotors are embedded with a function to determine the two nearest neighbors and virtually bond with them. Therefore, they move in two groups towards each other and towards the target in approaching phase. A collision is avoided and the strict formation are relaxes as perpetration to achieve the final formation.

FIGURE 6.28: 3D scenario-4 (a) initial config., (b) collision detection: perspective view, (c) collision detection: top view, (d) collision avoided, (e) spreading along d_G circle and (f) mission accomplishment
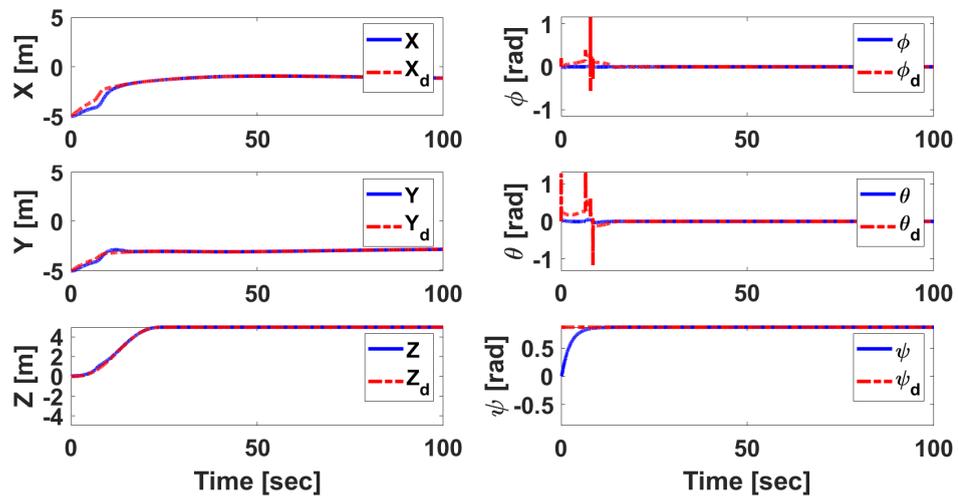
(a)
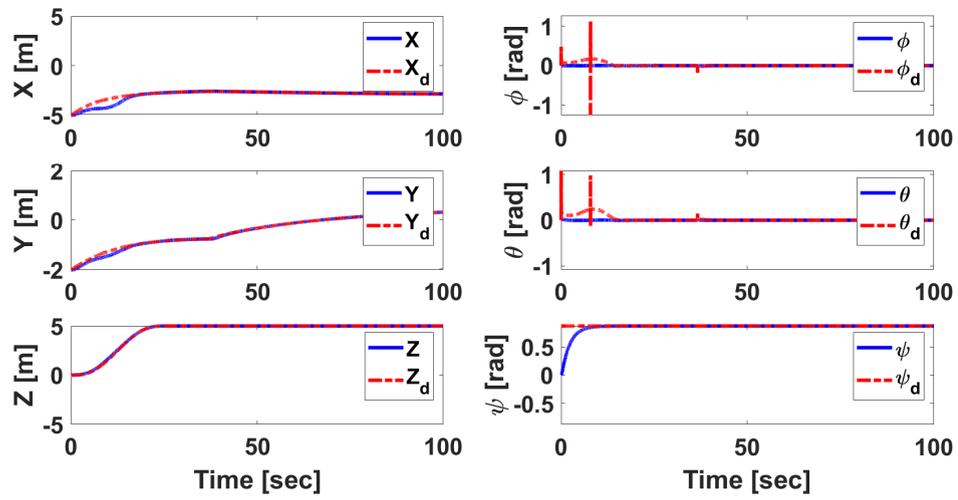
(b)
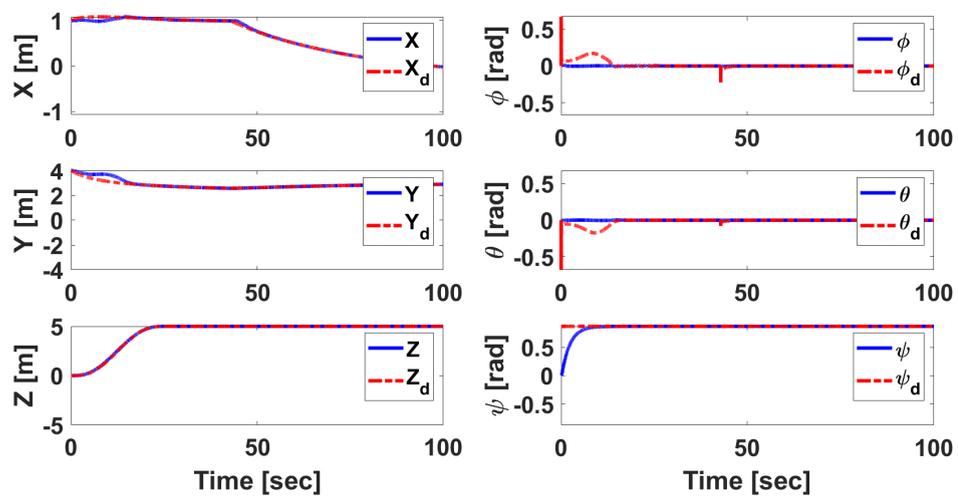
(c)

FIGURE 6.29: 3D scenario-4: position and orientation tracking performance (a) Q_1, (b) Q_2 and (c) Q_3
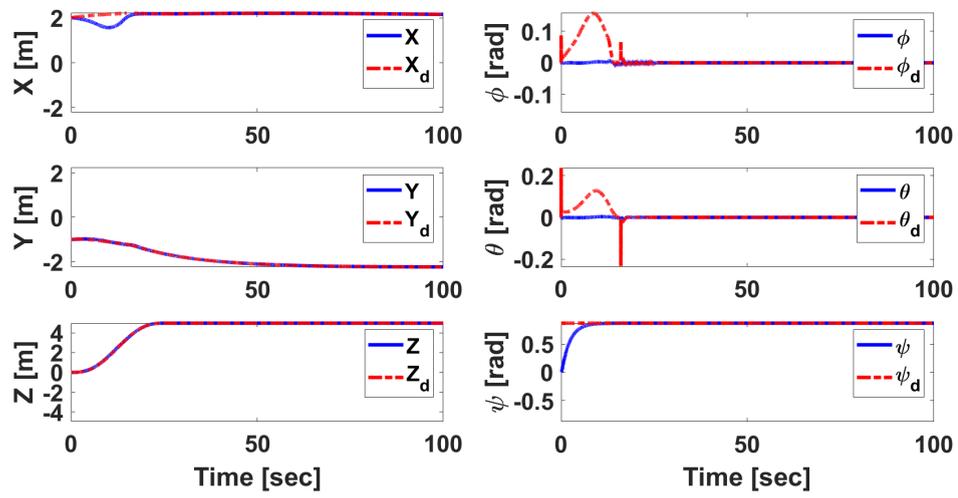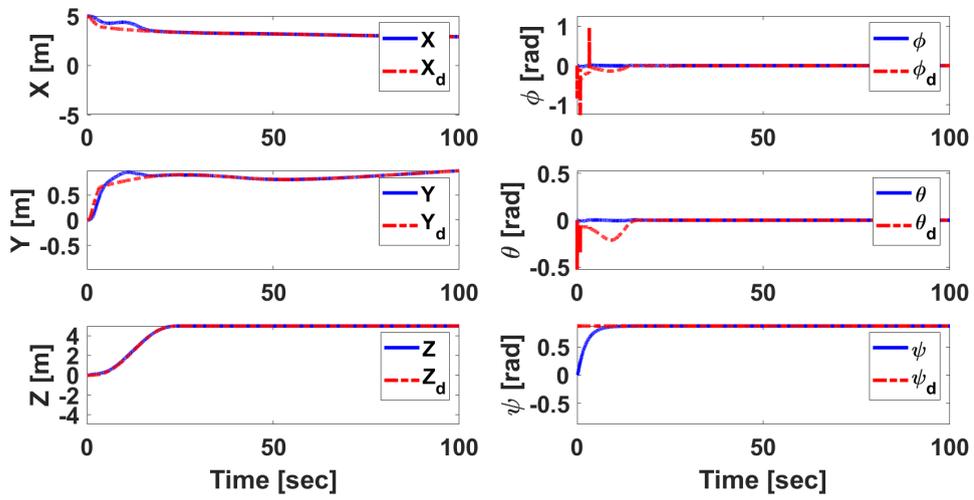
FIGURE 6.30: 3D scenario-4 (contd.): position and orientation tracking performance (a) Q_4 and (b) Q_5

Scenario 5

In the final scenario, a simulate a group of five quadrotors, three of which are clustered near each other 10 meters away from the goal.
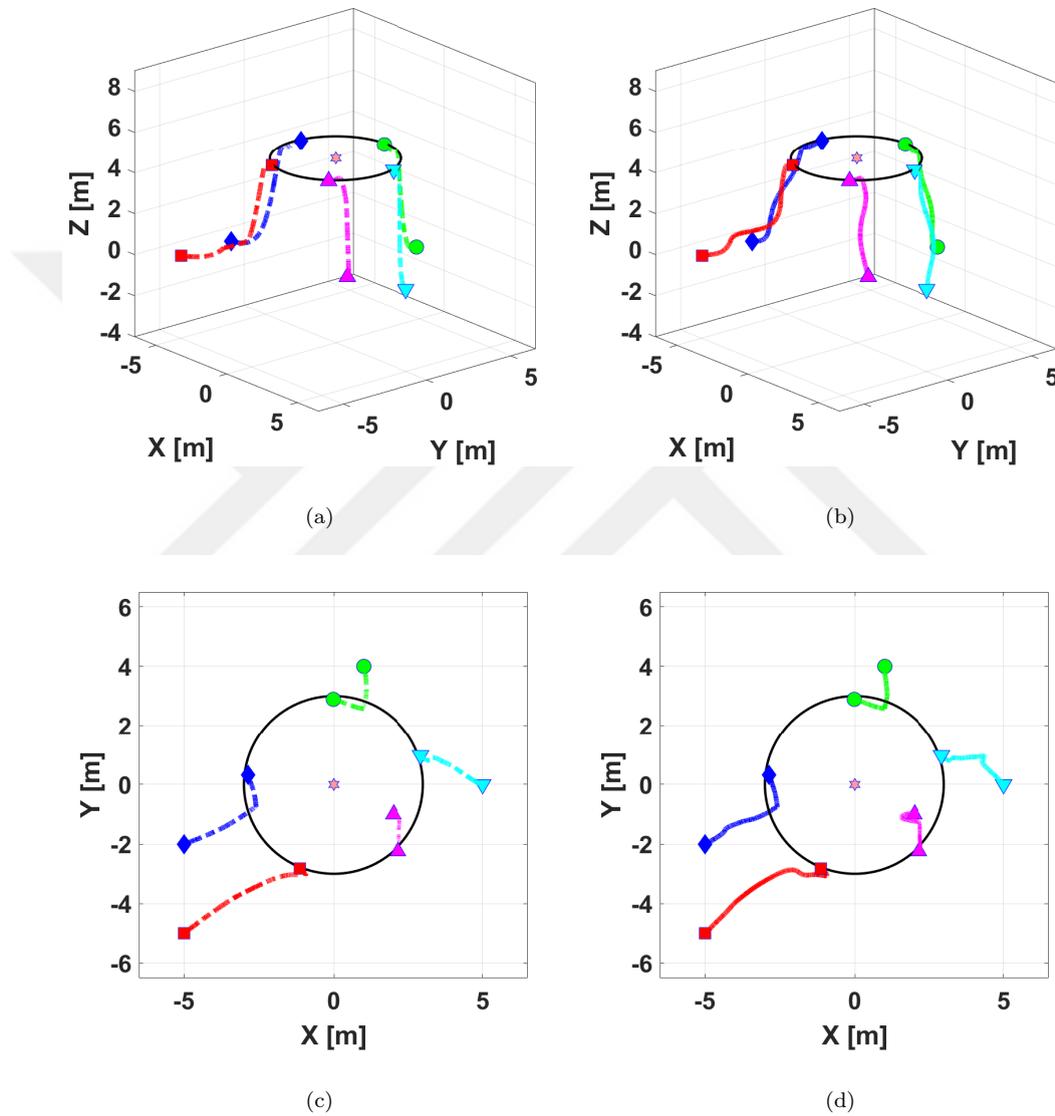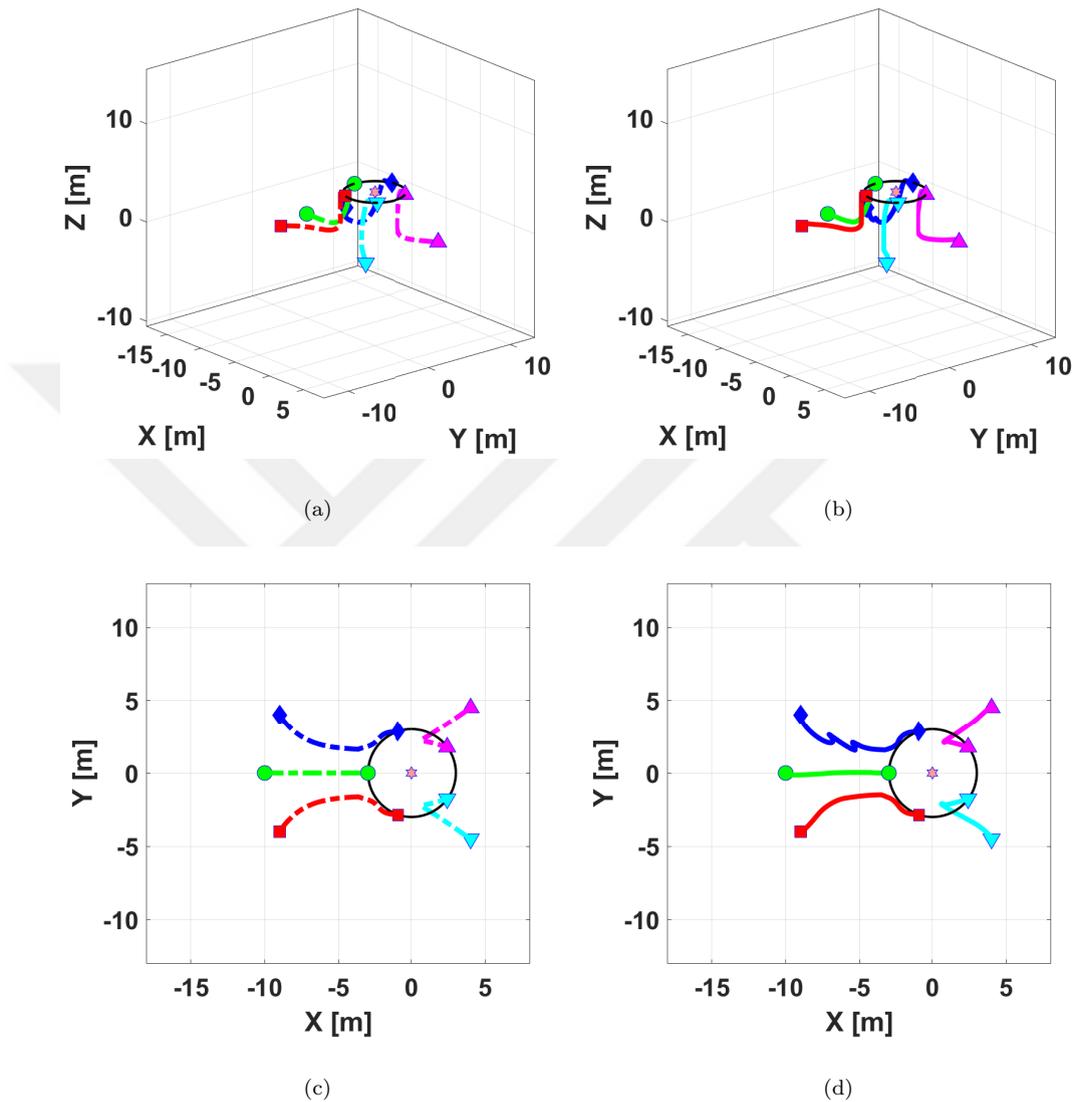

FIGURE 6.31: 3D scenario-5: (a) ref. traj., (b) actual traj., (c) ref. trajectories' projection on xy plane, (d) actual trajectories' projection on xy plane

The transition from the initial state of sparsely distanced quadrotors to formation state at a suitable execution time and compact formation size produces risk of collision, especially for robots with constraints and non linear dynamics. Quadrotors, in this scenario, follow their hyperbolic cylinder shaped motion profile but undergo a collision while doing so. Collision is then avoided successfully without negatively affecting the overall task.
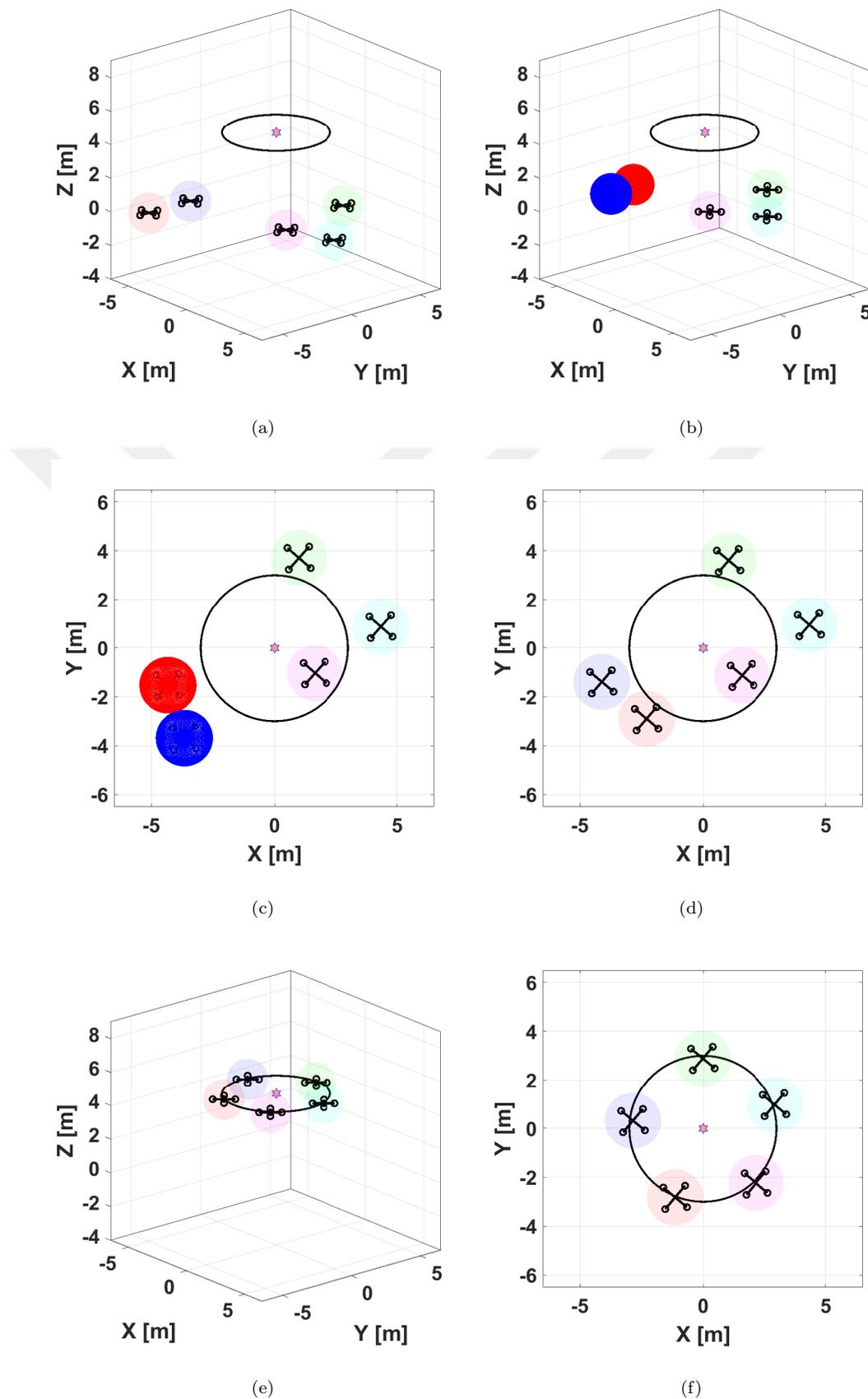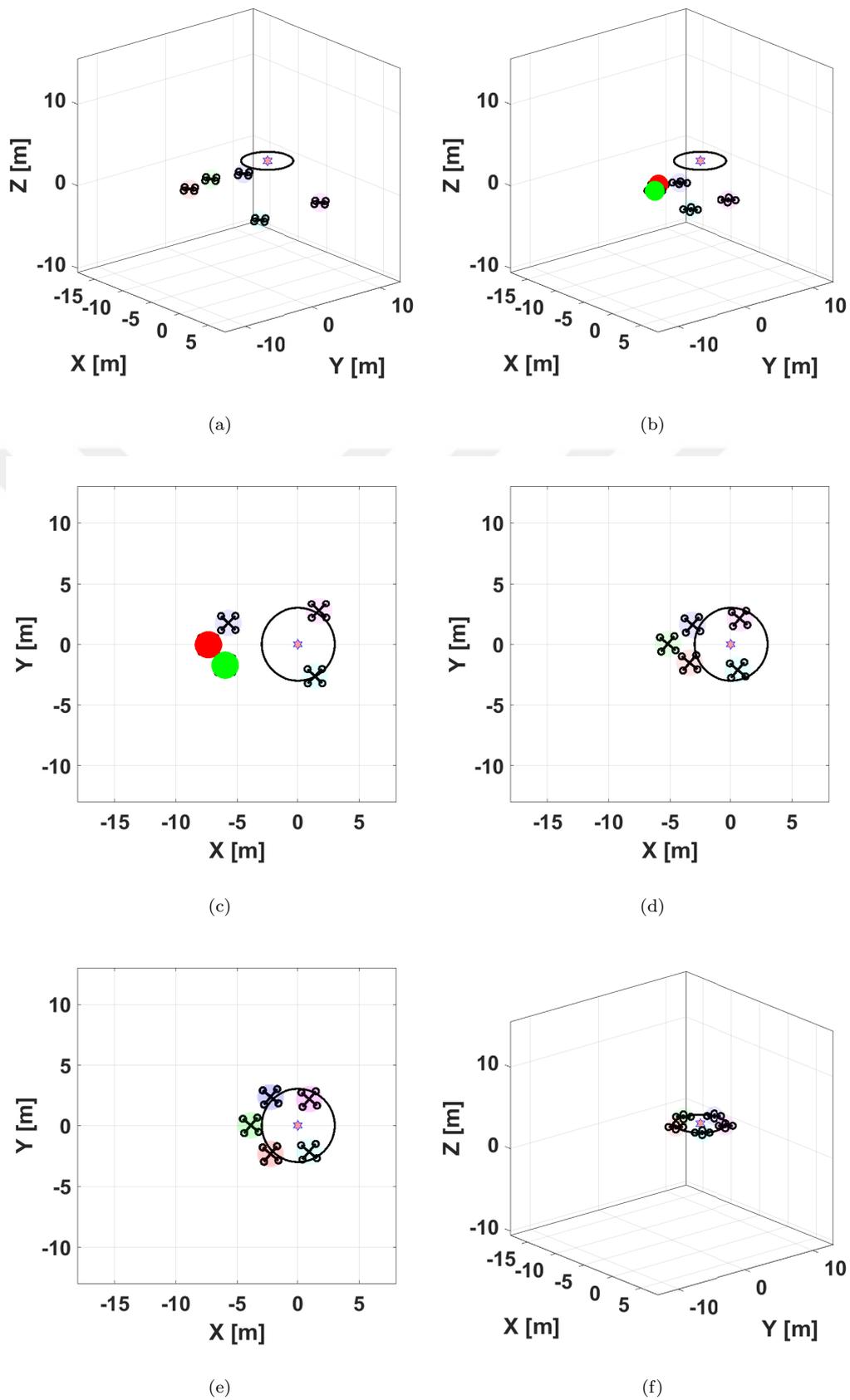
FIGURE 6.32: 3D scenario-5 (a) initial config., (b) collision detected: perspective view, (c) collision detected: top view, (d) collision avoided, (e) spreading along d_G circle and (f) mission accomplishment
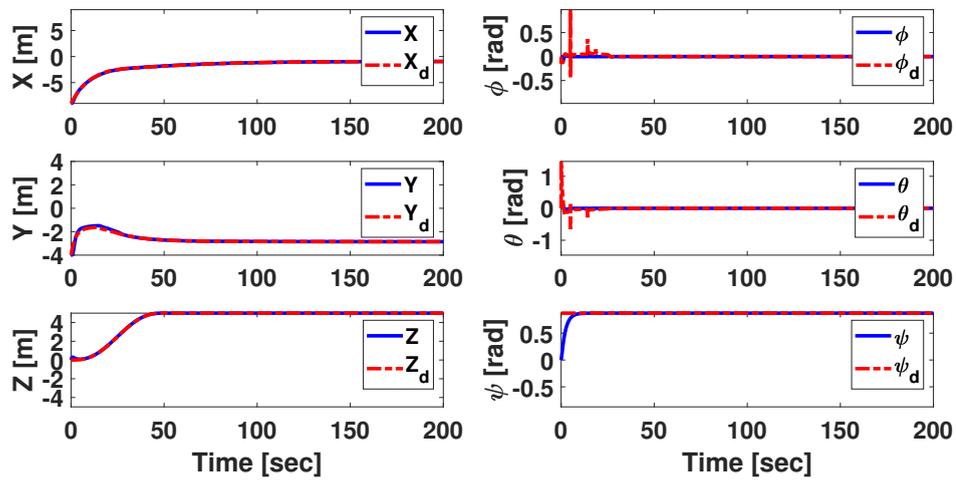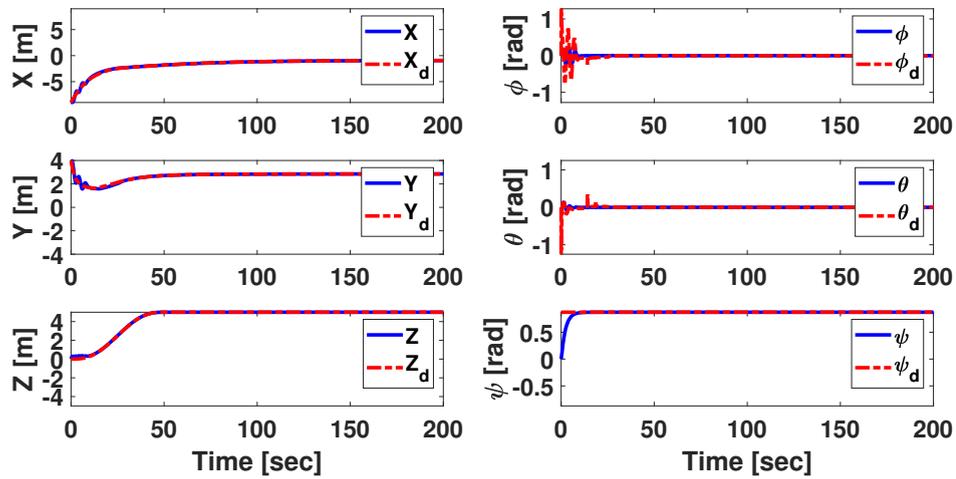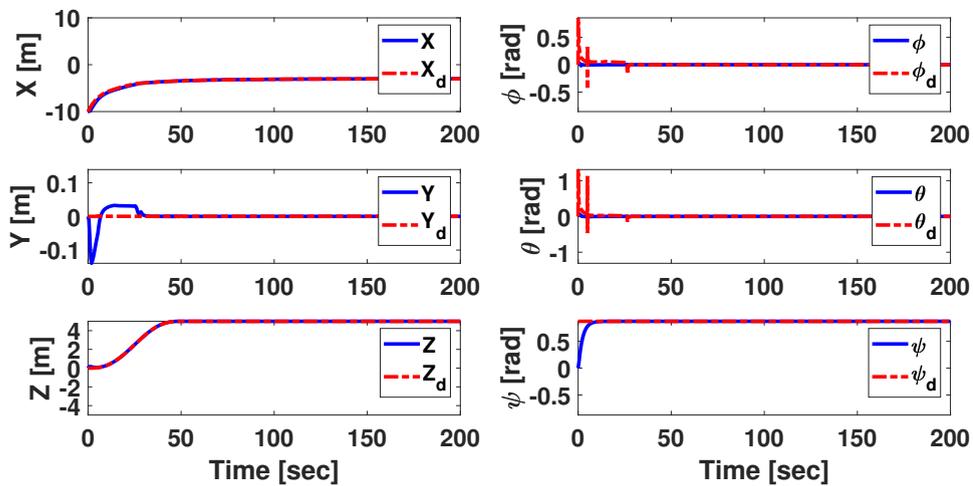
FIGURE 6.33: 3D scenario-5: position and orientation tracking performance (a) Q_1, (b) Q_2 and (c) Q_3
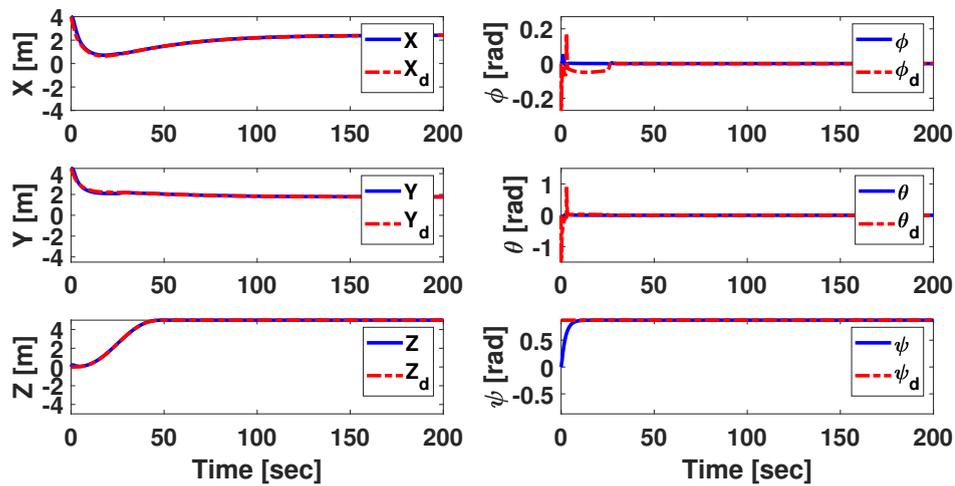
(a)
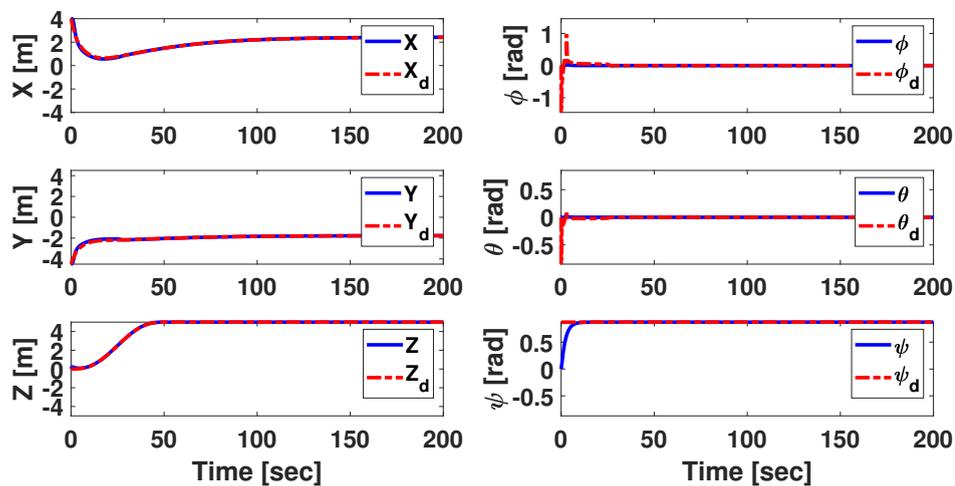
(b)

FIGURE 6.34: 3D scenario-5 (contd.): position and orientation tracking performance (a) Q_4 and (b) Q_5

To sum up, the proposed framework led to successful achievement of the formulated coordination problem in different scenarios and utilizing different dynamic robots in 2D and 3D. Sustained performance was witnessed for groups including more than three robots. Collision avoidance framework was able to limit the collisions to virtual shells thus safely avoid it for the physical robot. Single and multiple, sequential and parallel collision scenarios proved its robustness for average operating speed environments. Virtual inputs based control showed adequate performance under no external disturbances. Further concluding remarks are to be summarized in Chapter 7.

Chapter 7

Conclusions and Future Work

7.1 Concluding Remarks

In this thesis, a structure for MRS decentralized coordination has been developed. A novel online collision avoidance framework is proposed and integrated to the overall coordination structure. Both nonholonomic UGVs, and quadrotor type UAVs were taken into consideration as test beds for the framework. Several distinct scenarios were implemented in 2D and 3D simulation environments to inspect the functionality and performance of system's different blocks.

A high-level planner based on two layers of virtual viscoelastic bonds is introduced as a reference trajectory generation step. Together with the newly proposed collision avoidance algorithm, safe paths fulfilling coordinated motion sub-tasks defined in 1.1 are obtained. The framework was also extended to 3D workspace scenarios using polynomial trajectory generated altitude profile.

Desired trajectories tracking by physical robots was performed using model-based synthetic controllers. Virtual inputs approach based control is designed for both nonholonomic UGV and quadrotor UAV robots. Position and attitude controllers working in a hierarchical manner were implemented as the low-level motion control. Reference attitude angles are computed by utilizing the dynamic inversion method and they are used by the attitude controllers. Controller parameters were tuned

based on successive simulation results and their performance evaluation had been verified with simulations.

The collision avoidance framework, inspired by fish schools in nature, utilized the concept of virtual shells for collision detection and response. It allowed for safe collisions on the shell level so that the enveloped real robots' collisions are dodged. Two intuitive collision response algorithms were proposed and compared based on efficiency and communication criteria. Simulations verified that EC model, algorithm 1, is more efficient in terms of minimizing the change in the desired trajectory and thus worrying less about problems such as longer execution periods and unfeasible parts of the trajectories. On the other hand, LBR model, algorithm 2, needs less communication but on the expense of negatively affecting generated trajectories due to its *blindness* in scenarios including head-to-tail collisions and high operating velocities.

For 3D workspace implementation, generated trajectory smoothness is a major challenge. Trajectories were extended from horizontal plane derivation, which is based on holonomic point masses for abstraction. Thus, it was hard for the controllers to follow considering the nonlinear, coupled and underactuated nature of quadrotor UAVs. This problem was overcome by increasing viscous friction coefficient, thus slowing down the overall planned path together with similar logic parameter tuning.

The results of the simulations for the proposed framework with three, four, five and ten nonholonomic robots are all satisfactory in terms of task-oriented generated trajectories, controller performance and collision avoidance mechanism. Moreover, simulations for three and five quadrotor UAVs coordinated motion exhibited satisfactory results. In all scenarios, coordinated task was achieved. Simulations showed a consistent success of the collision avoidance algorithm across the increased number of robots in the system which is promising for the modular and autonomous nature of the system.

7.2   Future Work

While establishing different models and algorithms in this thesis, the focus was on the coordination structure as a whole and the particular problem of collision avoidance. However, an important factor such as communication got minor attention, as classified *out of scope*, and assumed to reliably exist. Efforts could be made on constructing communication protocols in favor of minimizing the bandwidth load of the system.

The collision avoidance scheme can be extended to static and dynamic obstacles. Shell's shape could be further optimized depending on robots' geometry as well. For instance, elliptical/ellipsoidal shells can be utilized to better represent non-holonomic robots while still preserving a high level of simplicity and abstraction in both collision detection and response. On the other hand, collision response algorithms, other than those proposed in this work, can be developed in the future.

Many parameters in the environment e.g. spring-damper coefficients, controller gain constants, were set based on simulation results in a trail-and-error manner. Learning-based techniques could further tune theses parameters for a satisfactory performance in different working scenarios.

Finally, as a natural extension of this work, we plan on the physical implementation of the proposed coordination scheme. Although different 2D and 3D robot models were covered, the proposed framework implementation could be extended to heterogeneous MRS. Different dynamical robot models can be included together with those already studied to furthermore investigate its robustness, e.g. biped robots, snake robots etc. Once implemented, tasks such as surveillance, border patrol, area coverage, visual shows, search, rescue and others can be performed with the aid of the proposed framework.

Bibliography

[1] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, pp. 345–383, Jun 2000.

[2] B. Kuipers, E. A. Feigenbaum, P. E. Hart, and N. J. Nilsson, "Shakey: From conception to history.," 2017.

[3] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent asimo: system overview and integration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2478–2483 vol.3, Sep. 2002.

[4] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10822 – 10825, 2008. 17th IFAC World Congress.

[5] J. Bohren, R. B. Rusu, E. Gil Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mösenlechner, W. Meeussen, and S. Holzer, "Towards autonomous robotic butlers: Lessons learned with the pr2," in *2011 IEEE International Conference on Robotics and Automation*, pp. 5568–5575, May 2011.

[6] N. Frietsch, O. Meister, C. Schlaile, and G. F. Trommer, "Teaming of an ugv with a vtol-uav in urban environments," in *2008 IEEE/ION Position, Location and Navigation Symposium*, pp. 1278–1285, May 2008.

[7] P. Caloud, Wonyun Choi, J. . Latombe, C. Le Pape, and M. Yim, "Indoor automation with many mobile robots," in *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, pp. 67–72 vol.1, July 1990.

[8] T. Fukuda, T. Ueyama, Y. Kawauchi, and F. Arai, "Concept of cellular robotic system (cebot) and basic strategies for its realization," *Computers and Electrical Engineering*, vol. 18, no. 1, pp. 11 – 39, 1992.

[9] S. C. Botelho and R. Alami, "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, pp. 1234–1239 vol.2, May 1999.

[10] Fang Tang and L. E. Parker, "Asymtre: Automated synthesis of multi-robot task solutions through software reconfiguration," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1501–1508, April 2005.

[11] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 476–481 vol.1, April 2000.

[12] J. Wawerla and R. T. Vaughan, "A fast and frugal method for team-task allocation in a multi-robot transportation system," in *2010 IEEE International Conference on Robotics and Automation*, pp. 1432–1437, May 2010.

[13] A. Prorok, A. Bahr, and A. Martinoli, "Low-cost collaborative localization for large-scale multi-robot systems," *2012 IEEE International Conference on Robotics and Automation*, pp. 4236–4241, 2012.

[14] Y. Hur, R. Fierro, and I. Lee, "Modeling distributed autonomous robots using charon: formation control case study," in *Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, 2003.*, pp. 93–96, May 2003.

[15] T. Rabie, A. Shalaby, B. Abdulhai, and A. El-Rabbany, "Mobile vision-based vehicle tracking and traffic control," in *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 13–18, Sep. 2002.

[16] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[17] J. Lin, A. S. Morse, and B. D. O. Anderson, "The multi-agent rendezvous problem," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 2, pp. 1508–1513 Vol.2, Dec 2003.

[18] S. G. Loizou, H. G. Tanner, V. Kumar, and K. J. Kyriakopoulos, "Closed loop motion planning and control for mobile robots in uncertain environments," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 3, pp. 2926–2931 Vol.3, Dec 2003.

[19] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 427–438, Feb 2013.

[20] R. Doriya, S. Mishra, and S. Gupta, "A brief survey and analysis of multi-robot communication and coordination," in *International Conference on Computing, Communication Automation*, pp. 1014–1021, May 2015.

[21] J. M. Hendrickx and V. Blondel, *Graphs and networks for the analysis of autonomous agent systems*. PhD thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2008.

[22] W. J. Dong, Y. Guo, and J. A. Farrell, "Formation control of nonholonomic mobile robots," *2006 American Control Conference*, pp. 6 pp.–, 2006.

[23] T. Azuma and T. Karube, "Formation control with fault-tolerance based on rigid graph theory," in *Proceedings of SICE Annual Conference 2010*, pp. 1137–1140, Aug 2010.

[24] A. Pierson and M. Schwager, "Controlling noncooperative herds with robotic herders," *IEEE Transactions on Robotics*, vol. 34, pp. 517–525, April 2018.

[25] A. A. Paranjape, S. Chung, K. Kim, and D. H. Shim, "Robotic herding of a flock of birds using an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 34, pp. 901–915, Aug 2018.

[26] X. Fang, Y. Wang, and Q. Lin, "Flight control of convertible uav based on extended-state-observer," 08 2011.

[27] Jinyan Shao, Guangming Xie, Junzhi Yu, and Long Wang, "Leader-following formation control of multiple mobile robots," in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pp. 808–813, June 2005.

[28] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013.

[29] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 813–825, Oct 2002.

[30] J. Ghommam, H. Mehrjerdi, and M. Saad, "Leader-follower formation control of nonholonomic robots with fuzzy logic based approach for obstacle avoidance," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2340–2345, Sep. 2011.

[31] Rongxin Cui, S. S. Ge, B. V. E. How, and Yoo Sang Choo, "Leader-follower formation control of underactuated auvs with leader position measurement," in *2009 IEEE International Conference on Robotics and Automation*, pp. 979–984, May 2009.

[32] T. Dierks and S. Jagannathan, "Neural network control of quadrotor uav formations," *2009 American Control Conference*, pp. 2990–2996, 2009.

[33] D. A. Mercado, R. Castro, and R. Lozano, "Quadrotors flight formation control using a leader-follower approach," in *2013 European Control Conference (ECC)*, pp. 3858–3863, July 2013.

[34] R. C. Arkin, *An Behavior-based Robotics.* Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[35] S. Monteiro and E. Bicho, "A dynamical systems approach to behavior-based formation control," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, pp. 2606–2611 vol.3, May 2002.

[36] Q. Jia and G. Li, "Formation control and obstacle avoidance algorithm of multiple autonomous underwater vehicles(auvs) based on potential function and behavior rules," in *2007 IEEE International Conference on Automation and Logistics*, pp. 569–573, Aug 2007.

[37] S. Rao and D. Ghose, "Sliding mode control-based autopilots for leaderless consensus of unmanned aerial vehicles," *IEEE Transactions on Control Systems Technology*, vol. 22, pp. 1964–1972, Sep. 2014.

[38] Weihua Sheng, Qingyan Yang, Jindong Tan, and Ning Xi, "Risk and efficiency: a distributed bidding algorithm for multi-robot coordination," in *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788)*, vol. 5, pp. 4671–4675 Vol.5, June 2004.

[39] J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar, and J. P. Ostrowski, "Cooperative localization and control for multi-robot manipulation," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 2, pp. 631–636 vol.2, Oct 2001.

[40] A. Borkowski, M. Gnatowski, and J. Malec, "Mobile robot cooperation in simple environments," in *Proceedings of the Second International Workshop on Robot Motion and Control. RoMoCo'01 (IEEE Cat. No.01EX535)*, pp. 109–114, Oct 2001.

[41] A. Jadbabaie, Jie Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, pp. 988–1001, June 2003.

[42] Wei Ren and R. W. Beard, "A decentralized scheme for spacecraft formation flying via the virtual structure approach," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 2, pp. 1746–1751, June 2003.

[43] T. H. A. van den Broek, N. van de Wouw, and H. Nijmeijer, "Formation control of unicycle mobile robots: a virtual structure approach," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 8328–8333, Dec 2009.

[44] E. Lalish, K. A. Morgansen, and T. Tsukamaki, "Formation tracking control using virtual structures and deconfliction," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 5699–5705, Dec 2006.

[45] D. Zhou, Z. Wang, and M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures," *IEEE Transactions on Robotics*, vol. 34, pp. 916–923, Aug 2018.

[46] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[47] J. S. Baras, Xiaobo Tan, and P. Hovareshti, "Decentralized control of autonomous vehicles," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 2, pp. 1532–1537 Vol.2, Dec 2003.

[48] S. Yannier, A. Sabanovic, and A. Onat, "Basic configuration for mobile robots," pp. 256 – 261 Vol.1, 01 2003.

[49] P. Vlantis, C. Vrohidis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Robot navigation in complex workspaces using harmonic maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1726–1731, May 2018.

[50] E. D. Rimon and D. E. Koditschek, "Exact robot navigatio n using artificial potential fields," 1992.

[51] W. Bernzen, "Active vibration control of flexible robots using virtual spring-damper systems," *Journal of Intelligent and Robotic Systems*, vol. 24, pp. 69–88, Jan 1999.

[52] J. T. Kim, J. Cho, B. Park, S. Chon, J. Seo, J. Kim, and S. Park, "Design of quaternion controller based on virtual spring-damper hypothesis for hydraulic manipulator," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 521–525, Aug 2016.

[53] P. A. M. M. B. Abeyrathna, W. A. S. P. Abeysiriwardhana, S. W. Amarasinghe, W. M. S. L. Ariyasinghe, and A. M. H. S. Abeykoon, "Simulation on active vibration suppression using virtual spring-damper combination," in *2013 International conference on Circuits, Controls and Communications (CCUBE)*, pp. 1–6, Dec 2013.

[54] N. Gulec and M. Unel, "A novel coordination scheme applied to nonholonomic mobile robots," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 5089–5094, Dec 2005.

[55] P. Urcola, L. Riazuelo, M. T. Lazaro, and L. Montano, "Cooperative navigation using environment compliant robot formations," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2789–2794, Sep. 2008.

[56] M. A. Guney and M. Unel, "Formation control of a group of micro aerial vehicles (mavs)," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 929–934, IEEE, 2013.

[57] M. Moore and J. Wilhelms, "Collision detection and response for computer animation," *SIGGRAPH Comput. Graph.*, vol. 22, pp. 289–298, June 1988.

[58] J. Wiech, V. A. Eremeyev, and I. Giorgio, "Virtual spring damper method for nonholonomic robotic swarm self-organization and leader following," *Continuum Mechanics and Thermodynamics*, vol. 30, pp. 1091–1102, Sep 2018.

[59] C. Richter, A. Bry, and N. G. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *ISRR*, 2013.

[60] D. Mellinger and V. S. A. Kumar, "Minimum snap trajectory generation and control for quadrotors," *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, 2011.

[61] Jinglai Shen, D. A. Schneider, and A. M. Bloch, "Controllability and motion planning of multibody systems with nonholonomic constraints," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 5, pp. 4369–4374 Vol.5, Dec 2003.

[62] C. Samson and K. Ait-Abderrahim, "Feedback stabilization of a nonholonomic wheeled mobile robot," in *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, pp. 1242–1247 vol.3, Nov 1991.

[63] D. Vissiere, D. E. Chang, and N. Petit, "Experiments of trajectory generation and obstacle avoidance for a ugv," in *2007 American Control Conference*, pp. 2828–2835, July 2007.

[64] K. Kyriakopoulos and G. Saridis, "Distance estimation and collision prediction for on-line robotic motion planning," *Automatica*, vol. 28, no. 2, pp. 389 – 394, 1992.

[65] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, pp. 417–422, Aug 2009.