UAV Based Group Coordination of UGVs

by

Soner Ulun

Submitted to the Graduate School of Sabancı University
in partial fulfillment of the requirements for the degree of
Master of Science

Sabancı University

July, 2013

UAV Based Group Coordination of UGVs

APPROVED BY:

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

Prof. Dr. Asif Sabanovic ....

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

Assoc. Prof. Dr. Mahmut F. Akşit ....

Asst. Prof. Dr. Hakan Erdoğan ....

DATE OF APPROVAL: 10/07/2013

UAV Based Group Coordination of UGVs

Soner Ulun

ME, Master's Thesis, 2013

Thesis Supervisor: Prof. Dr. Mustafa Ünel

Keywords: Autonomous Mobile Robot, UAV, UGV, Control, Coordination

Abstract

Coordination of autonomous mobile robots has received significant attention during the last two decades with the emergence of small, lightweight and low power embedded systems. Coordinated motion of heterogenous robots is important due to the fact that unique advantages of different robots might be combined to increase the overall task efficiency of the system.

In this thesis, a new coordination framework is developed for a heterogeneous robot system, composed of multiple Unmanned Ground Vehicles (UGVs) and an Unmanned Aerial Vehicle (UAV), that operates in an environment where individual robots work collaboratively in order to accomplish a predefined goal. UAV, a quadrotor, detects the target in the environment and provides a feasible trajectory from an initial configuration to a final target location. UGVs, a group of nonholonomic wheeled mobile robots, follow a virtual leader which is created as the projection of UAV's 3D position onto the horizontal plane. The UAV broadcasts its position at certain frequency to all UGVs. Two different coordination models are developed. In the dynamic coordination model, reference trajectories for each robot is generated from the motion of nodal masses located at each UGV and connected by virtual springs and dampers. Springs have adaptable parameters that allow the desired formation to be achieved In the kinematic coordination model, the position of the virtual leader and distances from the two closest neighbors are directly utilized to create linear and angular velocity references for each UGV. Several coordinated tasks are presented and the results are verified by simulations where different number of UGVs are employed and certain amount of communication delays between the vehicles are also considered. Simulation results are quite promising and form a basis for future experimental work on the topic.

Bir Grup İKA'nın İHA Tabanlı Koordinasyonu

Soner Ulun

ME, Master Tezi, 2013

Tez Danışmanı: Prof. Dr. Mustafa Ünel

Anahtar Kelimeler: Otonom Mobil Robot, İnsansız Hava Aracları, İnsansız Kara Aracları, Kontrol, Koordinasyon

Özet

Son yirmi yılda hafif ve az enerji tüketen gömülü sistemlerde yaşanmakta olan gelişmeler, otonom robot teknolojilerinde büyük gelişmelere yol açmıştır. Heterojen mobil robot gruplarının ilgi çekici olmasının en önemli sebeplerinden biri farklı tipteki robotların sahip olduğu avantajların tek bir sisteme entegre edilerek genel sistemin verimliliğinin arttırılabilmesidir.

Bu tez çalışmasında çoklu İnsansız Kara Araçlarından (İKA) ve bir adet İnsansız Hava Aracından (İHA) oluşan heterojen bir mobil robot grubunu belirlenmiş bir ortamda önceden verilen bir görevi tamamlamak için, her bir robotun iş birliği içinde çalışabileceği yeni bir teorik çerçeve geliştirilmiştir. İHA, dört pervaneli bir helikopter, ortamdaki hedefin konumunu hesaplayıp, herhangi bir başlangıç noktasından hedefe giden gerçekleştrilebilir bir yörünge çıkarabilmektedir. İKA'lar, holonomik olmayan iki tekerlekli mobil robotlar, sanal bir lideri takip etmektedirler; bu sanal lider ise İHA'nın üç boyutlu ortamdaki pozisyonunun yatay düzleme olan projeksiyonu olarak tanımlanmıştır. İHA kendi pozisyonunu belli aralıklar ile bütün İKA'lara göndermektedir. Bu tezde iki farklı koordinasyon modeli geliştirilmiştir. Dinamik kontrol modelinde, referans yörüngeleri nodal kütle, yay ve sönüm elemanı modeli kullanılarak her bir robot için ayrı ayrı oluşturulmaktadır. Kinematik koordinasyon modelinde ise sanal liderin pozisyonuna ve en yakın iki komşu robota olan uzaklıklar kullanılarak her bir robotun doğrusal ve açısal hız referans yörüngelerini hesaplanır. Çeşitli koordineli görevler sunulmuş ve bu çalışmalar benzetimlerle doğrulanmıştır. Benzetim çalışmaları sırasında robotlar arasında iletişimden kaynaklanan belli gecikmeler de göz önüne alınmıştır. Benzetim sonuçları oldukça umut vericidir ve sistemin deney düzeneği haline getirilmesi yönünde ilk adımı oluşturmaktadır.

Acknowledgements

I would like to extend my sincere gratitude to my thesis advisor Prof. Dr. Mustafa Ünel for his invaluable academic guidance, help and support, and sharing with me his scientific vision throughout my M.Sc study. His lectures and my discussions with him enabled me to focus on various aspects of autonomous mobile robots that will also be my main interest during my prospective PhD studies.

I also express my regards to my jurors; Prof. Dr. Asif Sabanovic, Assoc. Prof. Dr. Kemalettin Erbatur, Assoc. Prof. Dr. Mahmut F. Akşit and Assoc. Prof. Dr. Hakan Erdoğan for their valuable time and providing feedback on my M.Sc. thesis.

I would also be glad to acknowledge Dr. Gürdal Ertek's trust and patience with me through the last two year. I thank him for showing me the hardships and joys of an academician's life.

I like to thank to Control, Vision and Robotics Research Group team members; Bariş Can Üstündağ, Caner Şahin, Alper Yıldırım, Sanem Evren, Ibrahim Taygun Kekeç and Mehmet Ali Güney for being honest, resourceful and providing their precious help whenever I needed.

I send my sincere thanks to my colleagues Mine Saraç, Beste Bahçeçi, İlker Sevgen, Evrim Kurtoğlu, Giray Havur, Gökhan Göktürk and Ahmet Eren Demirel, and to my seniors Ahmet Fatih Tabak, Eray Baran and Edin Golubovic for being inspiration for me since my undergraduate studies.

Also I would like express my gratitude to the rest of the Mechatronics

Laboratory members for their friendship and the enjoyable moments that we have spent together.

Finally, I would like to thank my dear parents, my treasured siblings and extended family members for all their love, patience and the support throughout my life and my decisions. They were always there for me when I needed them while facing my good and bad choices in life.

Contents

List of Figures

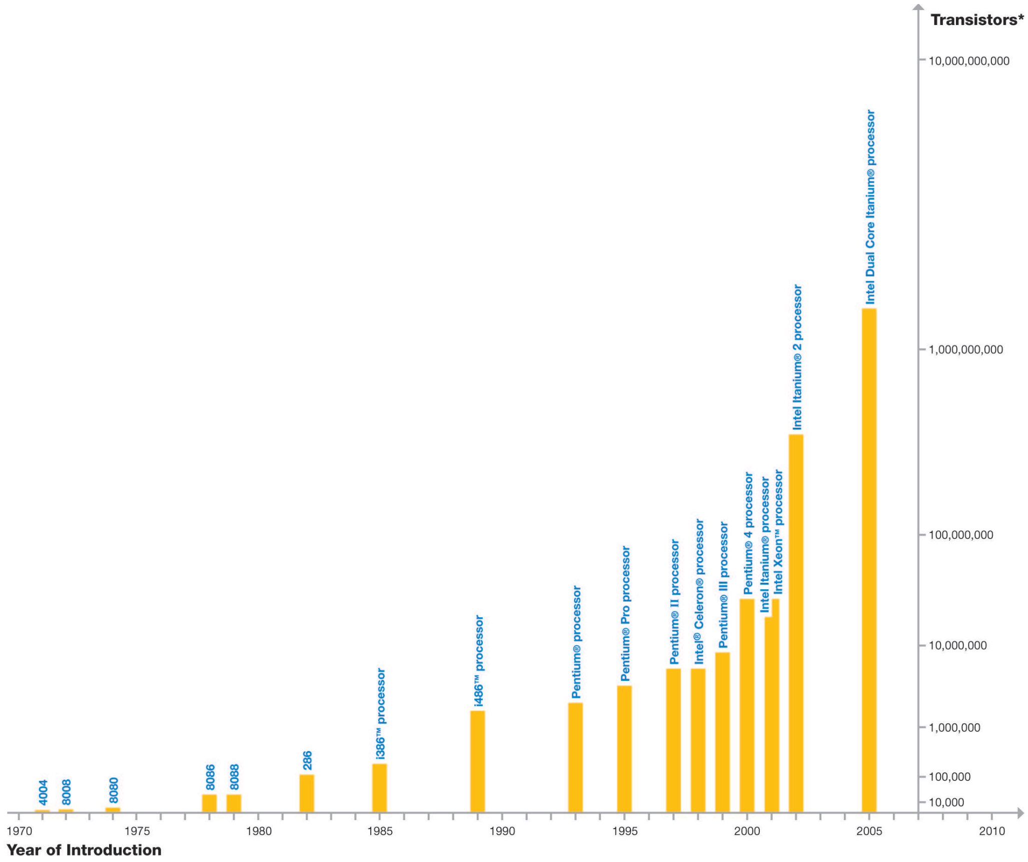
List of Tables

Chapter 1

Introduction

Robots are ideal to deal with the jobs that are undesired for humans. The importance of robotics has increased rapidly in the last five decades [1–6]. Robots have been utilized in an efficient manner in factories since 1961 [7]. Soon after that the interest towards mobile robots began to increase. Between 1966 and 1972, the first general-purpose mobile robot platform Shakey was built [8]. Due to the technology of its time, Shakey was expensive and very slow. The improvement on the electronics, sensors and low power microcomputers-microcontrollers units, had crucial effect on the development of mobile robots. In Figure 1.1 the development of the number of transistors on microprocessors over the last decades is depicted. Today mobile robots are becoming more reliable and affordable and faster every day with the improving technology.

The increasing computational capacity helped the improvement of the notion of autonomy in mobile robotics. Today mobile robots are capable of onboard computations that were not possible a decade ago. The importance of autonomous mobile robots has been increased as the demand for autonomous vehicles is increased. Also the works on mobile robots in different areas and applications increased greatly through the last two decades [9–14].



*Note: Vertical scale of chart not proportional to actual Transistor count.

Figure 1.1: The development of number of transistors on microprocessors during the last decades [15]

Autonomous mobile robots are suitable for the tasks that are dull, dirty and dangerous for humans. In order to accomplish such tasks, these robots need to adapt to unknown environments and dynamic workspaces which requires them to be more intelligent to be able to make their own decisions in such conditions. The environments that autonomous mobile robots are employed vary from constructed areas, like industrial plants to harsh and unconstructed dynamic environments such as in battlefields or in extraplanetary operations.

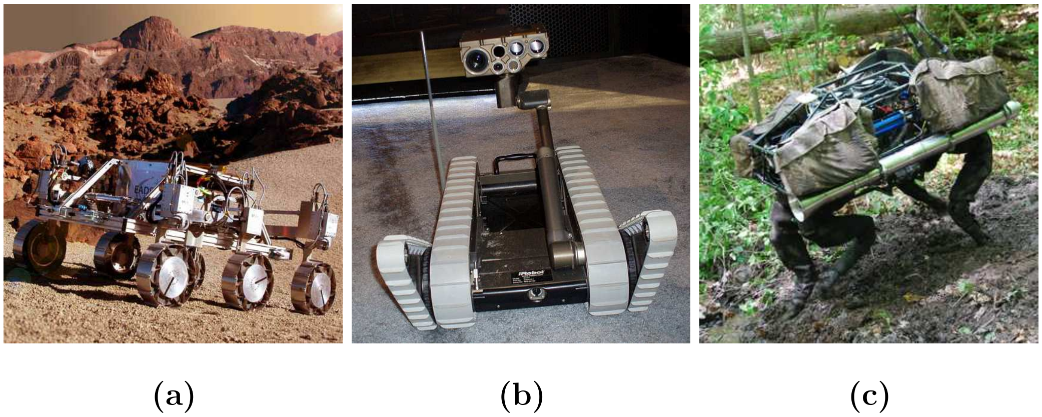
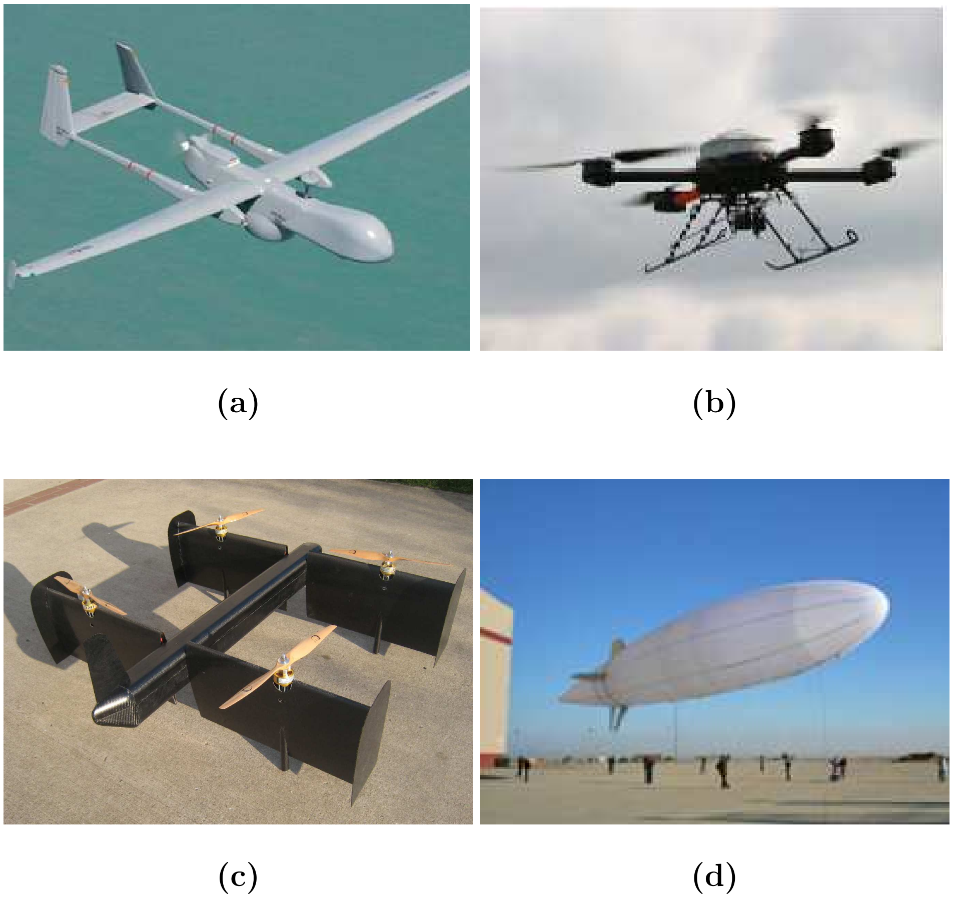


Figure 1.2: Common Unmanned Ground Vehicle types: (a) Wheeled UGV, (b) Tracked UGV, (c) Legged UGV.

The great variance in the working environments and applications brought out new issues for researchers to tackle. Unfortunately not all tasks can be accomplished with a single robot. This might be related with the medium that the autonomous mobile is in, its locomotion mechanism, constraints on its movements due to its design and so on. Instead of using a highly sophisticated, expensive robot which may require trained expert to operate, people are attracted to the idea of using cheaper and simpler robot groups that can accomplish same tasks on their own [16]. By doing so the risk of failure for a given task due to a malfunction is greatly reduced. Robot groups may consist of identical types of robots as well as different types of robots, thus forming a homogeneous or heterogeneous robot groups respectively.



(a)

(b)

(c)

(d)

Figure 1.3: Common Unmanned Aerial Vehicle types: (a) Fixed-wing UAV, (b) Rotary-wing UAV, (c) Tilt-wing UAV (d) Airship UAVs.

Unmanned robots can be categorized under three main groups; unmanned ground vehicles (UGVs), unmanned aerial vehicles (UAVs), and unmanned marine vehicles (UMVs). There are several different types under these categories. In Figure 1.2, Figure 1.3 and Figure 1.4 common types of unmanned ground vehicles, unmanned aerial vehicles and unmanned maritime vehicles are shown respectively.

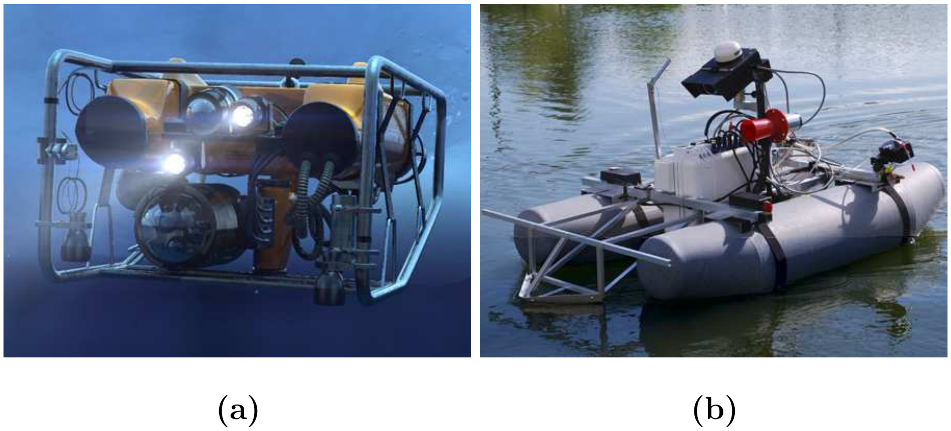


Figure 1.4: Common Unmanned Maritime Vehicle types: (a) Autonomous Underwater Vehicle, (b) Unmanned Surface vehicles.

The use of unmanned robots in industry can be seen in many ares such as; mining sites [17], agriculture, industrial facilities, warehouses, hospitals, sub-sea oil fields, pipelines. Some applications in those facilities can be given as follows: carrying products or a load from one location to another, structural analysis, fault detection, repairing tasks, wildlife detection, disinfection, mining and so on. Mobile robots can also be used for to civil activities such as fire detection, search and rescue of a survivor in natural disasters or accidents [12], nowadays they even entered to our houses as cleaning robots [18].

Unmanned robots group has also great importance in military applications such as searching mines [19], reconnaissance, crowd control missions [20] and as a security measure [21]. In [22] it is stated that the ultimate vision for an unmanned aerial vehicle in military applications is to be teamed with an unmanned ground vehicle over soil, with a unmanned marine vehicle over maritime environments, while being integrated with manned systems to improve and amplify the capabilities of the units.

Performing an intelligent task with a group of robots is still an important problem in academia. There are both advantages and disadvantages using a mobile robot group. The fundamental issues can be listed as:

- Communication,
- Data Fusion,
- Task allocation,
- Sensing and Path Planning,
- Dynamic obstacle avoidance.

These problems became more complicated in real life situations by considering the effects of resource failures, the possible distractive effects of robot on each other, inaccuracies in the plant models, time and energy constraints, computational costs, the presence of adversaries or changing goals according to the dynamical changes in the environment [23, 24]. Similarly, the general advantages of using robot groups can be emphasized as follows:

- Flexibility and robustness of the overall system is increased,
- Learning the environment in a faster manner,
- Decreasing the possibility of failures of mission due to malfunction of robots,
- Increased efficiency and solution quality.

In this thesis the emphasis is on the coordination scheme of a group of unmanned ground vehicles, nonholonomic mobile robots guided by an unmanned aerial vehicle. In literature there are several examples of UAV-UGV coordination and collaboration. The importance of this type of coordination comes from the capabilities of each vehicle. UAVs have greater visibility, and in most case they are in environments that the number of obstacles are scarce. However, power consumption and weight are nontrivial issues for flying robots, these facts have great impact on the equipment that can be used on the UAVs. On the other hand power consumption is less problematic for most UGVs, even if an UGV runs out of battery there is little risk of crushing or mechanical damage to the vehicle. Although the durability is a plus, the limited sight and stiffness are important drawback of UGVs. By using different types of mobile robots such shortcomings can be improved. A survey on previous works on UAV-UGV coordination can be found in Section 1.1, the following section after that outlines the thesis contribution and organization. Section underlines the works published during the context of this thesis. The last section in this chapter is Nomenclature.

1.1 Literature Survey on UAV-UGV Coordination

Mobile robot groups are capable of tasks where the task may not be accomplished by a single robot. In order to do so they rely on the capabilities of both theirs and the capabilities of the other robots in the group. For that reason coordination of different types of mobile robots can become beneficial. The development scheme for heterogenous mobile robot groups is an important and nontrivial issue. Researchers from diverse disciplines tackle this issue from various perspectives. In this section the importance of the cooperation schemes between UAV and UGV groups are described.

Previously different types of unmanned vehicles were mentioned. The UGVs can be classified under three main groups according to their locomotion method as follows:

- Wheeled UGVs
- Tracked UGVs,
- Legged UGVs

Each of these three main types have its own advantage and disadvantages. Wheeled robots are ideal for higher speed applications on environments with smooth surfaces. On the other hand, tracked UGVs are more preferable in rough terrain or on inclined surfaces and even climbing up and down the stairs. Lastly legged robots are more adaptable to both indoor and outdoor environments while the sophistication of the design process is higher.

A similar grouping can be cast for UAVs as follows:

- Fixed-wing UAVs,
- Rotary-wing UAVs,
- Tilt-wing UAVs,
- Airship UAVs

The advantage of fixed-wing UAVs is mainly low fuel consumption and thus long range. However these types of vehicles generally require runways to take off and land. Also these vehicles do not posses the capability to hover at any given position. On the other hand rotary-wing UAVs can hover at a point with ease and can perform takeoff and landing without a runway.

The main drawback for this type of vehicle is that they need to create a force greater or equal to their weight in order to keep their level of height. In other words they do not utilize the generated lift force using their wings, they constantly push the air downwards and form a force greater or equal to their weight. This action results with greater fuel consumption. On the other hand tilt-wing types have the abilities of both vehicle types in the expense of complex design and control. Finally the airship UAVs are suitable for tasks that require long periods, days instead of hours, and carrying large payloads, but they lack the speed and agility of other types of UAVs. It is crucial to underline that there are also various hybrid and unique designs for both UAVs and UGVs.These vehicles may not be included to any or can be included more than one of these groups.

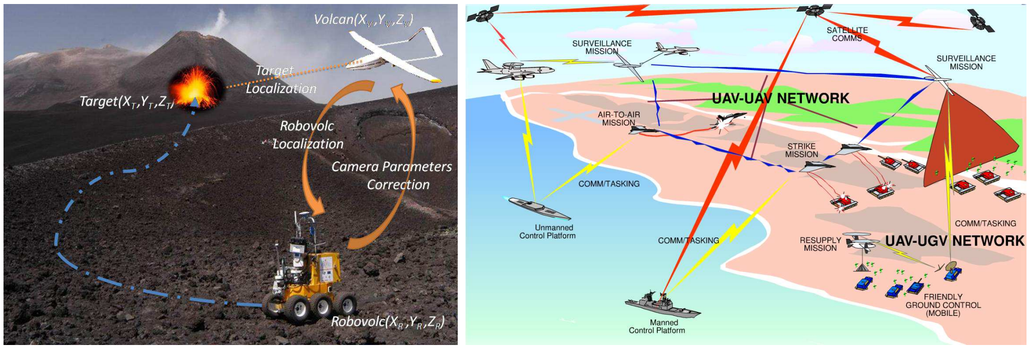
The physically beneficial relation between the robot groups constituting the robot group is a *marsupial relationship*. A marsupial relationship in the scope of a mobile robot system is defined as [25, 26] a relation of physical dependence of two or more mobile robots on each another, to diminish the individual weaknesses and overcome the tasks that they can not accomplish alone. This dependence can be for short durations or for long term. During this time robots can be dependent to others directives, power, communication or transport capabilities or any other physical superiority. These dependencies can be classified under roles such as; Carrier, Messenger, Coach, Manager, Processor, Supporter and Passenger robots [25]. This classification can be made in a more simpler manner for UAV-UGV cooperation. The following three types can be used to define the cooperative task accomplished by a heterogeneous mobile robot group [27].

- *Assistance from UAVs to UGVs*

- *Assistance from UGVs to UAVs*

- *Mutual cooperation of UAVs and UGVs*

Two possible coordination schemes are depicted in Fig.1.5. The details of the each cooperation scheme can be found in the following sections.



(a) **(b)**

Figure 1.5: Two heterogeneous coordination schemes: (a) An UAV and UGV coordination scheme [28], (b) A multiple heterogeneous robot groups coordination scheme [29].

1.1.1 Assistance from UAVs to UGVs

Aerial robots can assist ground robots in various aspects, mostly regarding to the information in the environment which can not be perceived by ground vehicles. This information can be the location of a target, location of robots in a group, map of the environment. Also UAV can provide as communication node between a remote operator and group.

In [30], the authors worked on the the problem of terrain classification to improve performance of the overall system. The image data from UAV is used to enhance the off-road and road performance of the UGV. By using aerial imagery, limited perception of UGV's on-board sensors are greatly increased. Infrared and RGB cameras mounted on an UAV provide the data required to classify the images. Aerial images from different terrains were classified by hand and then used to train and test the classifier. The trained classifier is able to generate a weight map. This map is used to detect the minimum energy path in the environment for UGV to a target position.

Luo et.al [31], designed a UAV-UGV team to coordinate in a GPS-denied environment, where the GPS information is lacking or can not be trusted. Their system consists of a quadrotor type UAV with horizontal and downward looking camera and two ground robots. Each robot is equipped with multiple sensors and on-board processors. Their control station is composed of two computers. A powerful portable PC with a wireless and a Bluetooth for image processing, controlling the robots and a workstation with internet communication to oversee the operation. Their scenario is to locate a victim in a hazardous zone, check the suspected location and then on the rescue mission of a possible victim. Their experimental results show that UAV is capable to search for a target in a suspected area using a preplanned path and send the precise target location back to the control station. UGV team is able to operate and find a target with the provided information, and then finish the rescue operation.

Similarly to [31], Frietsch et.al [16] employed an UAV to improved the navigation solution of a UGV. In their case the GPS loss is in a urban environment is due to noise, high buildings and so on. The pose of UAV is known and UAV is capable of detecting a tricolored pattern on the UGV. Using the location and direction of the UGV in each image and a geo-location algorithm the position and yaw angle of UGV is estimated with respect to a geo-coordinate frame. This information is then send to UGV to improve the navigation solution of the vehicle. Their experimental results showed that their proposed system is capable to produce similar results of the system with the GPS receiver and the magnetometer sensor on UGV.

1.1.2 Assistance from UGVs to UAVs

In this section the UAV-UGV cooperation is handled by the assistance of UGVs to UAVs in order to accomplish a given goal. These tasks might include but not limited to detect or provide landing areas for UAVs, transport the UAVs to a target, provide energy or to recover them after a mission.

In [32], the focus is on improving the endurance and range of a UGV with a gimballed landing for launching a helicopter type UAV. In that work the power consumption of each components are examined and battery technologies which are available for UGVs are surveyed and a gimballed landing, platform is designed. This platform enables an UAV, which is capable of vertical takeoff and landing, to be recharged and transported simultaneously. By using a group of UAV-UGVs, operational range can be maximized. The designs validity is checked with simulations.

Giakoumidis et.al. [33], presented a prototype pilot-scale system that was developed in their lab. The system, which is a small-scale indoor pilot version of a much larger outdoors full-scale system. The prototype system is composed of a UGV that serves as a recharge base and transporter for quadrotor UAV. UAV can be employed as a long range vision system for UGV. The authors also stated that algorithm can be adjusted to a large scale system with ease.

Mahoor et.al. [34] works on the problem of landing a light weight UAV to a smart landing platform using vision based techniques. This landing platform can be on top of a UGV as well as it can be used as a standalone module. Similar to previously mentioned works [32,33] this module can also be used for charging the UAV and increasing the flight range of the system. In order to detect the pose and successfully accomplishing the landing task, an onboard module is developed. This module is coupled with a attitude controller of the UAV and affect the landing process. The Scale Invariant Feature Transform (SIFT) algorithm is used for detecting the feature on the landing platform. RANSAC and Homography methods used to estimate the 3D orientation of the platform relative to UAV from detected features. The results of the proposed system is obtained via simulation.

In [35], the UGV is equipped with an interactive helipad and UGV acts as a transporter for the UAV. The UAV in this work is a small quadrotor equipped with horizontal and vertical cameras, which are used for inspecting the locations that can not be accessed with UGV and for landing respectively. According to the scenario the UAV-UGV team visits a specific stations periodically, the location of these stations are selected by qualified personnel. UGV continues through a predefined path. During this navigation the places of interest are scanned. For the locations that can not be reached by a UGV, UAV is launched to resume the inspection. A relative localization is achieved by using the known position of UGV during takeoff the UAV, then the desired place is scanned autonomusly. In order to accomplish such goal visual navigation, localization, visual landing and takeoff must be repeated many times. The landing task is eased with two newly proposed helipad designs. The first helipad is designed with tapered holes, where vertices of holes are located at the desired positions of landing gear of the UAV. The second proposed method is based on folding sideboards from the edges of the helipad actuated by a servo mechanism. To sum up the proposed coordinated scheme with two proposed helipad type is realized, and simulated. The errors during the tests showed that second helipad prototype is better.

1.1.3 Mutual cooperation of UAVs and UGVs

In this section the works where ground and aerial robots cooperate to achieve a task are described. The tasks can be achieved by jointly working of UAVs and UGVs in an UAV-UGV team. Basically any combination of the previous two assistance methods is grouped under this section.

In his work [36], Tanner stated that in near future intelligence, surveillance and reconnaissance (ISR) missions are going to be carried on by agile unmanned robots. In his work he developed a switched cooperative control architecture in order to locate a target that may or may not be moving in a given environment. A navigation function is integrated with the existing flocking algorithm. The shape control of UAV-UGV group is significantly improved and by employing provably convergent navigation functions. The UAVs are considered to fly at the same height and UGVs are considered to be omnidirectional. His proposed method is mainly centralized thus the communication and computational loads are increasing as the number of the robots increase. The efficiency of the approach is demonstrated in numerical simulations.

Grocholsky et.al. [37] developed a framework for UAVs-UGVsthat can be scaled and available for search and localization task. The trade of between UAVs-UGVs in terms of localization tasks are given as coverage and accuracy, while the coverage is superior in UAVs accuracy is limited, on the other hand UGVs can locate ground targets accurately. Their proposed framework is designed to be; easily implemented, independent of the number or the specificity of vehicles and offer guarantees for search and for localization. They also developed control algorithms and refined the quality of results by using estimators in order to solve the proposed the detection and the localization problems.

In [38], a hierarchical cooperative control framework for a UAV-UGV platform is proposed to detect and fight against wildfire. The emphasis of this work was on cooperative perception techniques instead of cooperative control of vehicles. They present a multi-layered hierarchical structure in this work. Upper layer is an airship acting as mission planner. The mission planner is responsible for system-level decision making and effective mission planning in a manner that optimizes time and resources of the system. The fire features are extracted from RGB and infrared images and a geo-located fire model constructed from these features. These features are also used for monitoring the dynamic fire evolution, and observing the performance of UAV-UGV team in fire suppression. The mission planner refines and updates the mission plan and reassigns the vehicles accordingly. For both UAVs and UGVs, there are separate mission planners, each planner tries to avoid collusion of each robot with other robots in its group. Finally a simulation platform to demonstrate the feasibility of the system involving an airship UAV, 2 quadrotor type UAVs and 2 wheeled UGVs is being developed.

During their effort to develop a heterogeneous and cooperative systems, Hager et.al [39] employed a group of stationary ground sensors. Their task is to help UAV-UGV group while performing an intelligence, surveillance, and reconnaissance (ISR) mission. Two UGVs are equipped with infrared sensor. The UAV facilitates a camera while stationary ground sensors are using radio frequency detectors. The motions of UGV is controlled by their cooperative autonomous system. The UGVs cooperate with the UAV and stationary ground sensors to verify the presence of a target, and then the information from all members the group is used to improve the estimate of target localization. To address the data fusion issue and non-deterministic latency of the radio, they developed the Out-Of-Order Sigma-Point Kalman Filter(O^3SPKF), which is a version of Sigma-Point Kalman Filter(SPKF). Their novel algorithm is tested in real world conditions.

1.2 Thesis Contributions and Organization

In this thesis two different coordination models are developed to solve the problem of navigating a group of UGVs, nonholonomic mobile robots, guided by an UAV, to accomplish a predefined goal. The location of the target position is known solely by the UAV. UGVs are guided to the vicinity of the goal position by following a virtual leader, which is obtained as the projection of UAV's 3D (x, y, z) position onto the horizontal plane, i.e. (x, y). UAV broadcasts its position information to the UGVs at certain frequency. The main difference between the two methods is the coordination logic for the UGVs. In the first method the UGV coordination is achieved by a simple dynamical model where virtual mass spring damper systems are utilized. In the second method, coordination between UGVs are achieved by defining appropriate linear and angular velocities using suitable kinematic relations. Results are successfully verified in simulation environment by a group of 3 and

5 UGVs, respectively. As UGVs, two wheeled nonholonomic mobile robots are selected. As a UAV, a quadrotor is chosen throughout the thesis. In these simulations, communication delays between robots are also considered, and for the second method obstacles are also included into the system.

The organization of this thesis is as follows: Chapter 2 is on the modelling and control of both nonholonomic mobile robot and a quadrotor type helicopter. In this thesis the UGVs are guided by a UAV toward a target while UGVs maintain a coordination between themselves. The first coordination is described in details in Chapter 3. The second coordination model is presented in Chapter 4. In Chapter 5 simulations results are given and discussions are provided for each simulation. Chapter 6 concludes the work with several remarks and indicates possible future directions.

1.3 Nomenclature

Symbol	Description
b	is the thrust factor of UAV
c_{neigh}	coefficient of virtual damper between two UGV
d_f	is the drag factor of UAV
d_{i2C}	signed distance of the closest UGV
d_{i2SC}	signed distance of the second closest UGV
d_{neigh}	desired distance between two UGV
e_f	the error of angles of UAV ϕ, θ and ψ
e_o	the error of position of UAV X, Y and Z
e_x	the error on x coordinate for reference tracking of mobile robot
e_{x_p}	is the error x coordinate for parking mobile robot
e_y	is the error y coordinate for reference tracking of mobile robot
e_{y_p}	is the error y coordinate for parking mobile robot
e_α	is the orientation error for reference tracking of mobile robot
e_{α_p}	is the orientation error for parking mobile robot
e_1, e_2, e_3	are the transformed errors for reference tracking of mobile robot
e_{1p}, e_{2p}, e_{3p}	are the transformed errors for parking of mobile robot
$E(\phi, \theta)$	angular velocity transformation matrix world to body frame
F_t	total force terms of the UAV
F_{i2C}	Forces due to the closest UGV
F_{i2SC}	Forces due to the second closest UGV
g	acceleration due to gravity
I_A	moment of inertia matrix of UAV in body frame
I_{XX}	moment of inertia of UAV around x-axis in body frame
I_{YY}	moment of inertia of UAV around y-axis in body frame
I_{ZZ}	moment of inertia of UAV around z-axis in body frame
J_P	is the polar moment of inertia of the propellers around the rotation axis

Symbol	Description
K_{d_f}	Derivative control gains for angles ϕ, θ and ψ
K_{d_o}	Derivative control gains for X, Y and Z
K_{i_f}	Integral control gains for angles ϕ, θ and ψ
K_{i_o}	Integral control gains for X, Y and Z
k_{p_1}, k_{p_2}	constant control gains for parking of the mobile robot
K_{p_f}	Proportional control gains for angles ϕ, θ and ψ
K_{p_o}	Proportional control gains for X, Y and Z
k_{neigh}	coefficient of virtual spring between two UGV
k_{t_1}, k_{t_2}	constant control gains for trajectory tracking of the mobile robot
L_A	is the distance from the center of the UAV to the center of the rotation axis of the propellers
L_G	is the half length of the wheel axis of the mobile robot
m	weight of the UAV
M_t	total moment terms of the UAV
p	angular velocity of the UAV along x-axis in body frame
q	angular velocity of the UAV along y-axis in body frame
r	angular velocity of the UAV along z-axis in body frame
${}^{W}\mathrm{R}_A$	Rotation matrix from body frame to the world frame
u_L	velocities of the centers of left wheel of the mobile robot
u_R	velocities of the centers of right wheel of the mobile robot
u_1	linear velocity of the mobile robot
u_1	linear velocity of the mobile robot
$u_{1_{ref}}$	linear velocity of the of the virtual reference robot
u_2	angular velocity of the mobile robot
$u_{2_{ref}}$	angular velocity of the virtual reference robot
v_{Ri}	the speed of R_i

Symbol	Description
v_{RiC}	velocities of the closest UGV
v_{RiSC}	velocities of the second closest UGV
v_x	linear velocity of the UAV along x-axis in body frame
v_y	linear velocity of the UAV along y-axis in body frame
v_z	linear velocity of the UAV along z-axis in body frame
V_A	linear velocities of the UAV in body frame
V_W	linear velocities of the UAV in world frame
x	x coordinate of the center of gravity for the mobile robot
X	x coordinate of the center of gravity for quadrotor
x_{ref}	x coordinate of the center of gravity for the mobile robot
y	y coordinate of the center of gravity for the mobile robot
Y	y coordinate of the center of gravity for quadrotor
y_{ref}	y coordinate of the center of gravity for the virtual reference robot
Z	z coordinate of the center of gravity for quadrotor
α	orientation of the mobile robot
α_{ref}	orientation of the virtual reference robot
$\omega_{1,2,3,4}$	propellers rotational speed
Ω_A	angular velocities of the UAV in body frame
Ω_A	angular velocities of the UAV in world frame
θ	pitch angle, angular position around y_w
θ_d	desired pitch angle
ϕ	roll angle, angular position around x_w
ϕ_d	desired roll angle
ψ	yaw angle, angular position around z_w
ψ_d	desired yaw angle

Chapter 2

Modeling and Control of UGVs and UAVs

In this chapter, we present background information on modeling and control of nonholonomic mobile robots and quadrotors.

2.1 Modeling & Control of Nonholonomic Mobile Robots

A nonholonomic robot is defined as a robot that has at least one nonholonomic kinematic constraints. A nonholonomic mobile kinematic constraint can be described based on the relationship between the differential degrees of freedom of the mobile robot and the degrees of freedom in the workspace [40]. If the differential degrees of freedom is less than the degrees of freedom in the workspace then the mobile robot has a kinematic constraint, thus it is a nonholonomic mobile robot. In other words if a vehicle has constraints on its velocity while moving in certain directions then it is called a nonholonomic mobile robot. Cars, motorbikes and bicycles are the most common examples of nonholonomic vehicles in our daily life. All these vehicles lack the ability to move sideways.

Unfortunately nonholonomic constraints complicates the development of control law and mathematical model for the system, since it requires dealing with rank deficient systems or so called underactuated systems. In this work even though being aware of these complications we are working with an *unicycle* type of two wheeled mobile robots. This is mainly because of the mechanical complexity of realization of the holonomic systems. Also in literature there is a great interest in nonholonomic mobile robots, especially during the last two decades [9, 16, 31, 41–43]. From here on the unicycle mobile robot is going to be mentioned as UGV.

2.1.1 Modeling

As previously mentioned our specific UGV introduces a constraint on its velocity. Due to this constraint the speed to the sideways directions are zero, thus the robot is not able to perform sideways motion under no-slip or no-skid assumptions. The well studied kinematic model for unicycle robot is given by the equation [43]:

$$\begin{aligned} \dot{x} &= u_1 \cos\alpha \\ \dot{y} &= u_1 \sin\alpha \\ \dot{\alpha} &= u_2, \end{aligned} \tag{2.1}$$

where x and y represents the Cartesian coordinates of the center of gravity for the mobile robot. α is the orientation of UGV with respect to the horizontal axis. A UGV is shown in Figure 2.1. The linear and angular velocities in (2.1), u_1 and u_2, are dependent to the velocities of the centers of right and left wheels, u_R and u_L respectively. This relation can be shown as:

$$\begin{aligned} u_1 &= \frac{1}{2}(u_R + u_L) \\ u_2 &= \frac{1}{2L_G}(u_R - u_L)\ , \end{aligned} \tag{2.2}$$

and L_G is the half length of the wheel axis as shown in Figure 2.1.

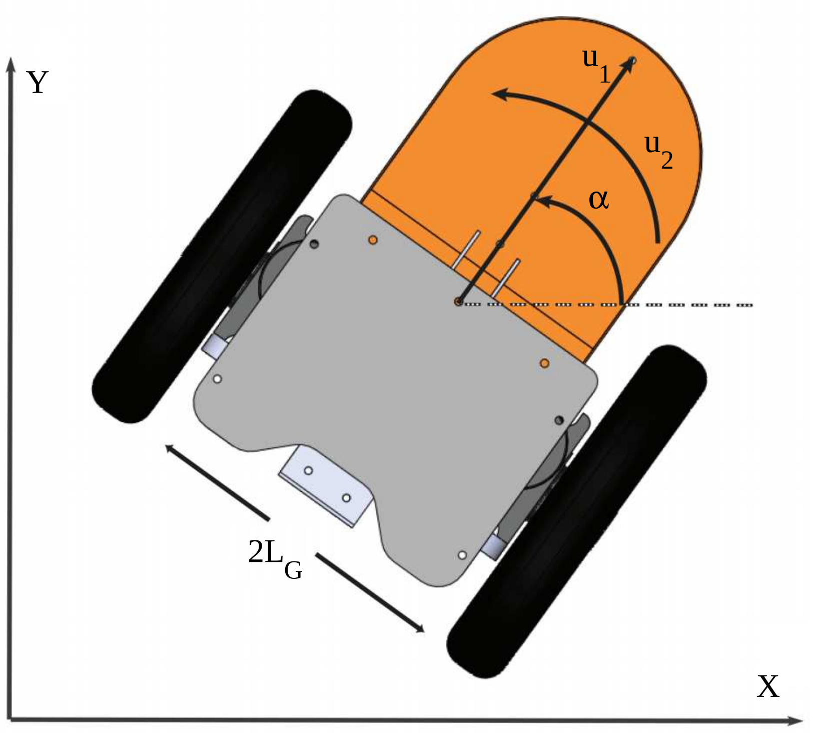


Figure 2.1: A unicycle robot and its variables of interest.

The equation 2.1 can be rewritten in the following form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos\alpha \\ \sin\alpha \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2 \ . \tag{2.3}$$

In equation 2.3 , the variables x, y and α can be treated as outputs, while u_1 and u_2 can be introduced as inputs for controlling the pose of the UGV. This equation set simply tells us that in order to control pose and orientation of the UGV on the horizontal plane linear and angular velocities of the robot should be designed appropriately.

2.1.2 Control

The control of a UGV is a complicated tasks due to the fact that it is an underactuated system. UGV has three outputs and only two inputs to control. The control of a UGV is discussed for two different cases which are trajectory tracking and parking. In order to explain this problem we are taking the second derivative of the outputs:

$$\begin{aligned} \ddot{x} &= \dot{u}_1 cos\alpha - u_1 u_2 sin\alpha \\ \ddot{y} &= \dot{u}_1 sin\alpha + u_1 u_2 cos\alpha \end{aligned} \tag{2.4}$$

by using simple algebra we can rewrite the equation as follows:

$$\begin{aligned} \dot{u}_1 &= \ddot{x} cos\alpha + \ddot{y} sin\alpha \\ u_2 &= \frac{1}{u_1}(\ddot{y} cos\alpha - \ddot{x} sin\alpha) \end{aligned} \tag{2.5}$$

As it can be seen in equation 2.5 the linear velocity and the angular velocity are coupled. There is a singularity when the linear velocity is zero, due to this fact the parking of UGV needs to be tackled by utilizing other methods. In literature there are several methods to park a UGV. These methods are mostly based on feedback stabilization, fuzzy logic, optimal control and Lyapunov type of methods [43–46]. One important fact is the switching frequency between the controllers, high frequency switching might lead to instability.

Trajectory Tracking

In order to make the UGV successfully follow a time varying reference trajectory, a *virtual reference robot* is put in use [47]. The idea behind this approach is to create reference trajectory that abides the nonholonomic constraint, which limits the speed of the robot in sideways directions. The model used for *virtual reference robot* can be found below:

$$\begin{bmatrix} \dot{x}_{ref} \\ \dot{y}_{ref} \\ \dot{\alpha}_{ref} \end{bmatrix} = \begin{bmatrix} \cos\alpha_{ref} \\ \sin\alpha_{ref} \\ 0 \end{bmatrix} u_{1ref} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_{2ref} \ . \tag{2.6}$$

where x_{ref} and y_{ref} represents the Cartesian coordinates of the center of gravity for the virtual reference robot. α_{ref} is the orientation of virtual reference robot with respect to the horizontal axis. The controls $u_{1_{ref}}$ and u_{2ref} are the linear and angular velocities of virtual reference robot.

By designing x_{ref}, y_{ref} and α_{ref} continuously differentiable and bounded when time goes to infinity, it can be shown that:

$$\begin{bmatrix} u_{1_{ref}} \\ u_{2_{ref}} \\ \alpha_{ref} \end{bmatrix} = \begin{bmatrix} \dot{x}_{ref} \cos \alpha_{ref} + \dot{y}_{ref} \sin \alpha_{ref} \\ \left(\ddot{y}_{ref} \dot{x}_{ref} - \ddot{x}_{ref} \dot{y}_{ref} \right) / \left(\dot{x}_{ref}^2 + \dot{y}_{ref}^2 \right) \\ \arctan \left(\dot{y}_{ref} / \dot{x}_{ref} \right) \end{bmatrix} . \tag{2.7}$$

The tracking errors e_x, e_y and e_α are defined as the difference between the pose of the actual robot and the pose of the virtual robot as follows:

$$\begin{bmatrix} e_x \\ e_y \\ e_\alpha \end{bmatrix} = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} - \begin{bmatrix} x_{ref} \\ y_{ref} \\ \alpha_{ref} \end{bmatrix} . \tag{2.8}$$

The transformed tracking errors e_1, e_2 and e_3 are defined by using an invertible transformation as follows:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\alpha \end{bmatrix} . \tag{2.9}$$

The transformation used on the tracking error is a Rotation matrix, which has a determinant of one and its inverse is equal to its transpose.

$$\begin{bmatrix} e_x \\ e_y \\ e_\alpha \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} . \tag{2.10}$$

By using this fact it can be concluded that the norm of $\begin{bmatrix} e_x & e_y & e_\alpha \end{bmatrix}^T$

is equal to the norm of $\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T$, thus the transformed errors are bounded if and only if the tracking errors are also bounded. In [48] Samson shows that using the transformed errors, the following controls is able to regulate the tracking errors to zero for time varying trajectories.

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -k_{t_1} e_1 + u_{1ref} \cos e_3 \\ -u_{1_{ref}} \left(\dfrac{\sin e_3}{e_3} \right) - k_{t_2} e_3 + u_{2_{ref}} \end{bmatrix}. \tag{2.11}$$

where k_{t_1} and k_{t_2} are positive constant control gains.

Parking Problem

As previously explained parking the robot at a fixed reference pose is different than trajectory following. Similar to the trajectory tracking error, the parking error is defined as follows:

$$\begin{bmatrix} e_{x_p} \\ e_{y_p} \\ e_{\alpha_p} \end{bmatrix} = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} - \begin{bmatrix} x_p \\ y_p \\ \alpha_p \end{bmatrix}. \tag{2.12}$$

where e_{x_p}, e_{y_p} and e_{α_p} are the fixed reference position and orientation respectively. A similar transformation as given in (2.9) can be applied to obtain transformed parking errors (e_{1p}, e_{2p} and e_{3p}) for easier construction of the control law as follows:

$$\begin{bmatrix} e_{1p} \\ e_{2p} \\ e_{3p} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{x_p} \\ e_{y_p} \\ e_{\alpha_p} \end{bmatrix}. \qquad (2.13)$$

In [48] it is shown that using the transformed errors, the following controls is able to regulate the tracking errors to zero for fixed references.

$$\begin{bmatrix} u_{1p} \\ u_{2p} \end{bmatrix} = \begin{bmatrix} -k_{p_1} e_{1p} \\ -k_{p_2} e_{3p} + e_{2p}^2 \sin(t) \end{bmatrix}. \qquad (2.14)$$

where k_{p_1} and k_{p_2} are positive constant control gains. For both trajectory tracking and parking, the control gains are determined by extensive simulation by using Matlab V7.14 and Simulink V7.9.

2.2 Modeling and Control of a Quadrotor

Quadrotor is a rotary-wing vehicle with four motors. Main features of rotary-wing vehicles are their agility, maneuverability, being the capable to hover around a point in empty space. Another strategic advantage of this type of vehicle is its vertical take of capability. Because of these features, unlike most fixed-wing vehicle, rotary-wing vehicles does not require runways and capable of working both indoors and outdoors. Helicopter and quadrotors are the most commonly used rotary-wing vehicle in daily life.Especially during the last decade quadrotors became very popular, there are various work in literature for both utilizing the abilities of quadrotors and the control problem due to its nature [16, 49–51].

Similar to UGV, unicycle mobile robot, quadrotors are also underactuated systems. There are four control inputs,from four motors of the quadrotor, while there are six variables to control. X, Y, and Z are the center of mass of quadrotor in world frame. ϕ, θ, ψ are the quadrotors attitude in world coordinate system.In this work from here on a quadrotor is going to be mentioned as UAV.

In Figure 2.2 the coordinate axes, moments and torques exerted on the UAV is depicted. The world frame is denoted with $^{W}(.)$, while the frame attached to the center of gravity of the UAV is denoted with $^{A}(.)$. The orientation of the body frame with respect to the world frame is expressed with a rotation matrix in 2.15,

$$^{W}\mathrm{R}_A(\phi,\theta,\psi) = \begin{bmatrix} c\psi c\theta & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\psi c\theta & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{2.15}$$

$c(.)$ and $s(.)$ denotes $cos(.)$ and $sin(.)$, respectively.

Transpose of the same transformation matrix in 2.15 is also used in order to transform velocities from the body of the UAV to the world frame as follow:

$$V_A = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = {}^{W}\mathrm{R}_A^T(\phi,\theta,\psi) \cdot V_W \tag{2.16}$$

Similarly, transformation relation from time derivative of the attitude angles $\begin{bmatrix} \dot\phi & \dot\theta & \dot\psi \end{bmatrix}^T$ to the angular velocities of the vehicle $\begin{bmatrix} p & q & r \end{bmatrix}^T$ is defined as:

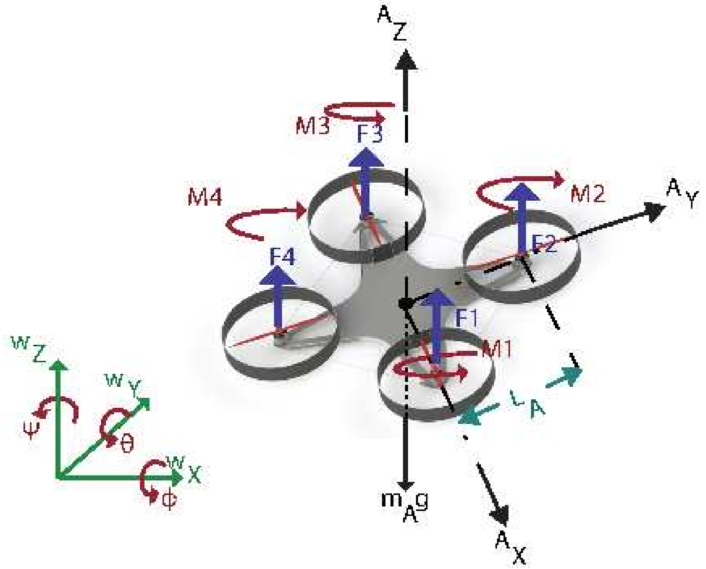


Figure 2.2: A quadrotor with exerted moments and forces.

$$\Omega_A = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = E(\phi, \theta) \cdot \Omega_W \tag{2.17}$$

where E is the angular velocity transformation matrix and defined as:

$$E(\phi, \theta) = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \tag{2.18}$$

2.2.1 Modelling

As previously mentioned modelling of a quadrotor is well studied. To derive the dynamic model of a UAV Newton-Euler formulation is a preferable method. In order to do so the aerial vehicle is assumed to be a 6 DOF rigid body. The dynamics of UAV can be formed as:

$$\begin{bmatrix} mI_{3x3} & 0_{3x3} \\ 0_{3x3} & I_A \end{bmatrix} \begin{bmatrix} \dot{V}_W \\ \dot{\Omega}_A \end{bmatrix} + \begin{bmatrix} 0 \\ \Omega_A \times (I_A \Omega_A) \end{bmatrix} = \begin{bmatrix} F_t \\ M_t \end{bmatrix} \tag{2.19}$$

m is the mass and I_A denotes the vehicle's inertia matrix expressed in body frame. I_A can be represented with a three by three diagonal matrix with elements, I_{XX}, I_{YY} and I_{ZZ}, the moments of inertia about X, Y, and Z axes, respectively. I_{3x3} is used for representing a three by three identity matrix while 0_{3x3} matrix indicates a three by three zero matrix. The left hand side of Eqn.(2.19) is standard for many aerial vehicle types.Unfortunatelly the right hand side terms, total force and moment terms, F_t and M_t, differ for various aerial vehicle types. The derivation of the Newton-Euler model of a quadrotor is not going to be carried out in this work since it is out of the scope. Further information on the following UAV model please refer to [49, 51].

$$\begin{aligned} \ddot{X} &= (cos\phi sin\theta cos\psi + sin\phi sin\psi)\frac{U_Z}{m}, \\ \ddot{Y} &= (cos\phi sin\theta sin\psi - sin\phi cos\psi)\frac{U_Z}{m}, \\ \ddot{Z} &= (cos\phi cos\theta)\frac{U_Z}{m} - g, \end{aligned} \tag{2.20}$$

$$
\begin{aligned}
\ddot{\phi} &= \frac{I_{YY} - I_{ZZ}}{I_{XX}}\dot{\theta}\dot{\psi} + \frac{J_P}{I_{XX}}\Omega + \frac{U_\phi}{I_{XX}}, \\
\ddot{\theta} &= \frac{I_{ZZ} - I_{XX}}{I_{YY}}\dot{\phi}\dot{\psi} + \frac{J_P}{I_{YY}}\Omega + \frac{U_\theta}{I_{YY}}, \\
\ddot{\psi} &= \frac{I_{XX} - I_{YY}}{I_{ZZ}}\dot{\phi}\dot{\theta} + \frac{U_\psi}{I_{ZZ}}
\end{aligned}
\tag{2.21}
$$

In the equations above J_P is the polar moment of inertia of the propellers around the rotation axis. The virtual control inputs of the UAV and the relation between the motor speeds, ω_1, ω_2, ω_3 and ω_4, are defined as:

$$
\begin{aligned}
U_Z &= b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2), \\
U_\phi &= L_A b(\omega_2^2 - \omega_4^2), \\
U_\theta &= L_A b(\omega_3^2 - \omega_1^2), \\
U_\psi &= d_f(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2), \\
\Omega &= -\omega_1 + \omega_2 - \omega_3 + \omega_4
\end{aligned}
\tag{2.22}
$$

The coefficients b is thrust factor, d_f is the drag factor and L_A is the distance from the center of the UAV to the center of the rotation axis of the propellers.

It is crucial to note that the Eqn.(2.21) is valid for hover or very small values of the angles ϕ, θ and ψ. It is known that using small angle approximation $sin\vartheta \approxeq \vartheta$ and $cos\vartheta \approxeq 1$. As ϕ, θ and ψ goes to zero, using this identity on Eqn.(2.18) it can be shown that:

$$E(\phi,\theta) = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \approxeq I \tag{2.23}$$

Since the angles are very small or the UAV is in hover mode it is safe to assume that the derivatives of the ϕ, θ and ψ are zero. Then by taking derivative of Eqn.(2.17) it can be shown that:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \dot{E} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + I \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \approx \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \tag{2.24}$$

2.2.2 Control

In order to control the UAV we are using a cascaded control scheme. The inner controller stabilizes the orientation angles in order to achieve a stable flight while the outer controller is responsible for the control of the position of UAV.

Attitude Control of UAV

Assuming ϕ, θ and ψ are small, we can linearize attitude dynamics given in (2.21) as follows:

$$\ddot{\phi} = \frac{U_\phi}{I_{XX}}, \quad \ddot{\theta} = \frac{U_\theta}{I_{YY}}, \quad \ddot{\psi} = \frac{U_\psi}{I_{ZZ}} \tag{2.25}$$

The control inputs U_ϕ, U_θ and U_ψ are regulated by using a PID controller for each attitude angle. The mathematical formulation of the PID controller and errors can be seen in Eqn.(2.26).

$$\begin{aligned} e_f(t) &= f_{des} - f(t) \\ U_f(t) &= K_{p_f} e_f(t) + K_{i_f} \int_0^t e_f(\tau) d\tau + K_{d_f} \frac{de_f(t)}{d(t)} \end{aligned} \tag{2.26}$$

where $f =\phi$, θ and ψ. K_{p_f}, K_{i_f} and K_{d_f} are the tuning parameters for the weights of proportional, integral and derivative control respectively for PID controllers. K_{p_f} affects the response of the system directly, the bandwidth of the controller is determined mainly by this component. High K_{p_f} gains might lead to high oscillations as well as instability in the overall system while low gains result with unresponsive systems. K_{i_f} value varies with the integral of the error, main effect is to eliminate the steady-state error. Unfortunately inappropriate K_{i_f} gains might easily lead to oscillations. The last component K_{d_f} varies with the time derivative of the error, a damping effect is introduce with this component while it slows the response of the system.

Position Control of UAV

The position controller is responsible for generating the desired attitude reference values of the attitude controller while following a feasible trajectory. The position controller is designed by using (2.20).

$$\begin{aligned} \ddot{X} &= (cos\phi sin\theta cos\psi + sin\phi sin\psi)\frac{U_Z}{m}, \\ \ddot{Y} &= (cos\phi sin\theta sin\psi - sin\phi cos\psi)\frac{U_Z}{m}, \\ \ddot{Z} &= (cos\phi cos\theta)\frac{U_Z}{m} - g, \end{aligned} \tag{2.27}$$

These accelerations are required in order to follow a desired trajectory. In order to regulate the second derivatives of X, Y and Z, virtual control inputs are using PID controllers as follows:

$$\mu_o = K_{p_o} e_o + K_{io} \int_0^t e_o dt + K_{do} \dot{e}_o \tag{2.28}$$

where O =X, Y and Z. By equating the second derivatives of X, Y and Z to the virtual controls inputs for the position controller the following equations are obtained:

$$\begin{aligned} \mu_X &= (cos\phi sin\theta cos\psi + sin\phi sin\psi)\frac{U_Z}{m}, \\ \mu_Y &= (cos\phi sin\theta sin\psi - sin\phi cos\psi)\frac{U_Z}{m}, \\ \mu_Z &= (cos\phi cos\theta)\frac{U_Z}{m} - g \end{aligned} \tag{2.29}$$

The virtual control inputs μ_X, μ_Y, μ_Z can be used to compute U_Z, ϕ_d, θ_d as in 2.30. The key assumption is that the desired yaw angle, ψ_d is constant during the flight [52].

$$\begin{aligned} U_Z &= m\sqrt{\mu_x^2 + \mu_y^2 + (\mu_z + g)^2}, \\ \phi_d &= sin^{-1}\left(\frac{m(\mu_x sin\psi_d - \mu_y cos\psi_d)}{U_1}\right), \\ \theta_d &= tan^{-1}\left(\frac{\mu_x cos\psi_d + \mu_y sin\psi_d}{\mu_z + g}\right) \end{aligned} \tag{2.30}$$

Calculated ϕ_d and θ_d and the constant ψ_d are the reference inputs for the attitude controller. It is important to note that ψ_d can be chosen arbitrarily inside the range of $[0\ \pi)$.

Chapter 3

A Simple Dynamic Coordination Model

In Chapter 1 the importance of coordinated motion and different types of coordination schemes were described. To move a group of autonomous robots in a coordinated manner, each robot should affect one or more robots, and this affect can be reflected to the generated position and orientation references of the robots.

This chapter describes the details of the first coordinated motion algorithm between UAV and UGVs. The coordination of UGVs are based on the defined forces modeled with mass spring and damper system based on [42]. The details of the improved algorithm is underlined. Also the trajectory reference used for the UAV is described in detail.

In order to establish the coordination between UGVs using forces and the dynamics a *virtual reference model* is designed. This reference model considers the interaction of each robot, R_i, with two closest robots and the virtual leader, V_L. In order to improve the efficiency of the system the spring coefficient is selected as a smoothly varying function with respect to the distance to V_L. The regulation of the position and orientation errors between the virtual reference robot's pose and the actual robot's pose to zero are guaranteed with a smooth time-varying feedback control law. The projection of the UAV's position onto the X-Y plane is defined to be the position of the virtual leader V_L.

3.1 Formulation of Coordinated Task

A group of UGVs, n nonholonomic mobile robots, namely $R_1, R_2, \ldots, R_{n-1}, R_n$, and a quadrotor type aerial vehicle, UAV, are considered to accomplish a coordinated task of navigating from some initial configuration to a target location denoted by T. The target and all of the mobile robots are assumed to be on the same plane.

A scenario is considered where the success of the coordinated task is determined by accomplishing several objectives. These objectives are:

- UAV is able to locate T from a distance and can hover on the target T after reaching it.
- $R_1, R_2, \ldots, R_{n-1}, R_n$ should surround the target, T, and form a circle with radius d_T where T is located at the center.
- The UGVs should be evenly spaced on the circle.
- The orientation of each R_i should be towards T once the previous three objectives are successfully accomplished.

The nature of each task can be different from one another, but in order to accomplish these tasks together, each R_i might check if all the other UGVs have achieved the same state before starting the next phase. This objective needs to be realized via wireless communication, after acquiring desired position and/or orientation. Collision avoidance is an essential problem in the context of coordinated motion [11]. In each phase UGVs should avoid collisions with each other or any obstacle in the environment. Initially we assume that there are no obstacles on the plane of UGVs. The UAV is assumed to be able to detect obstacles and avoid collisions by generating an appropriate trajectory. The obstacles are included on the plane of UGVs after the success at the obstacle-free environment.

In this study, we assume a stationary target, T, position of which is unknown to the UGVs but known to UAV. Detection task can be accomplished using visual features on T, with or without a priori knowledge about the target. UGVs are not allowed to park till UAV hovers on T. When UAV is on top of T, it broadcasts a signal that informs UGVs to park when they are equally spaced from target and from their closest neighbours. In the absence of this signal each R_i tries to surround the position coordinates that is currently known as the position of T. The UAV broadcasts its position on the horizontal plane as the position of T, till it reaches on top of actual T. UGVs are guided by UAV to the vicinity of actual T without being affected from the initial distance to the target.

Another assumption in our work is that each R_i, is capable of perceiving its environment by some appropriate sensor, in order to find if there is any object in their *virtual collision prediction region* [53]. Figure 3.1 depicts the proposed framework.

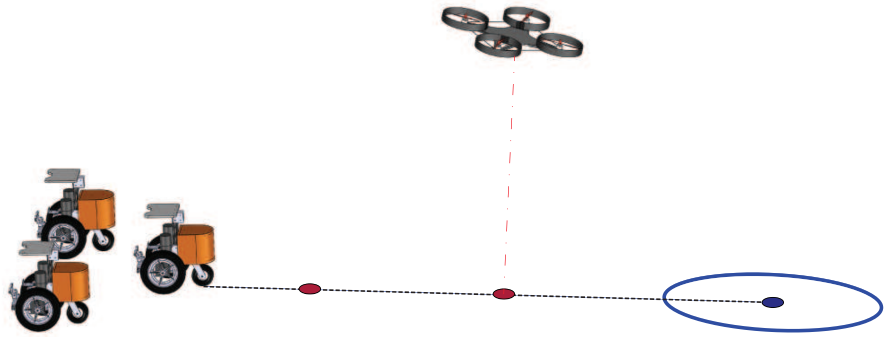


Figure 3.1: Depiction of coordination scheme, where UAV, UGVs, T and UAV's projection onto horizontal plane are indicated.

3.2 Virtual Reference System for UGV

The virtual reference system of robots can be modeled using numerous different methods. Nonetheless the system should be keep as simple as possible for computational efficiency the better design. Electrostatic forces and gravitational forces and spring mass damper can be given as an example for such system. In this section a spring mass damper system with adaptive spring constant is used to coordinate the UGVs.

The proposed method consist of virtual masses equal to the number of UGVs, n, in order to generate reference trajectories for each UGV. Every virtual mass, $(m_{V1}, m_{V2}, ...m_{Vi}..., m_{Vn})$, is a point particle with a finite mass. Since the virtual reference robots are point particles their orientation is not defined. The virtual masses are connected to each other, thus reference trajectory for each UGV is generated in a collective manner. For each mVi the dynamics are imposed by using the interaction due to neighbours and attraction to the virtual leader.

As it is discussed in Chapter 2 UGVs are not particle, they have constraints on their velocity in sideways direction. It is important to note that by using the point particle for generating references we are also relaxing the nonholonomic constraint on UGVs. As a result generated references can move towards any direction, where the orientation is calculated using reference velocities.

Using inspiration from nature, the number of neighbours affecting coordination is limited to two. In other words each R_i is affected by its closest neighbour, R_iC, second closest neighbour, R_iSC as well as the V_L. In Fig. 3.2 such scenario is depicted with four UGV. As it can be seen in the figure the fourth UGV is not related with the course of R_i. Also the spring mass damper system between the UGVs and V_L is depicted.

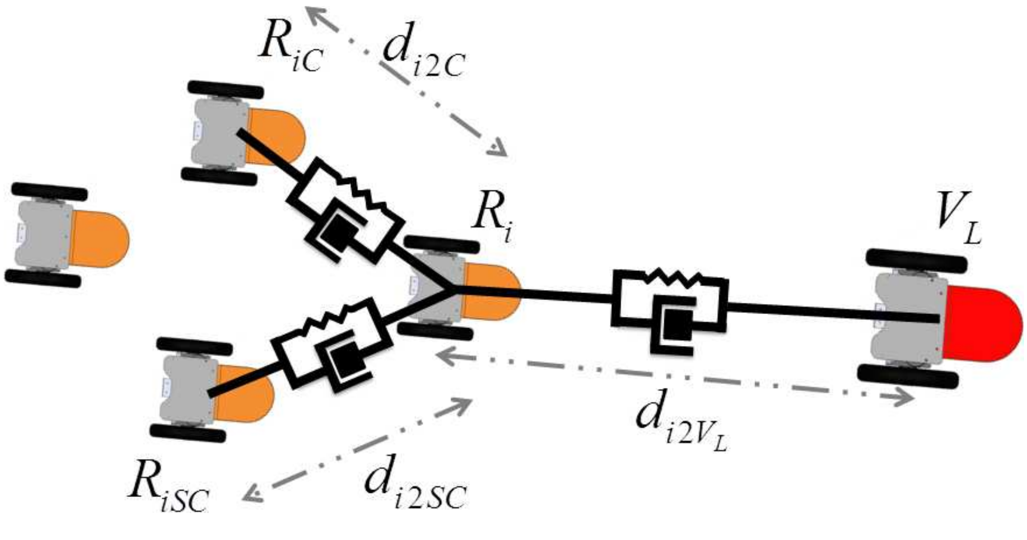


Figure 3.2: Closest two neighbors and V_L, with spring mass damper systems that defines the coordination force

d_i2C, is the signed distance of the closest UGV from R_i and d_i2C, is the signed distance of the second closest UGV from R_i. d_iV_L, is the signed distance of R_i from V_L. R_iC and R_iSC tries to maintain a certain distance, the distances d_i2C and d_i2SC are used for calculating the force due to neighbours, F_{neigh} when this equilibrium distance is not met. This behavior is exactly same as elongating and shrinking of an spring to its equilibrium length.

The effecting force of the neighbours R_iC and R_iSC on R_i for coordination is defined as follows:

$$\begin{aligned} F_{i2C} &= -[k_{neigh}(d_{i2C} - d_{neigh}) + c_{neigh}(v_{Ri} - v_{RiC})]\overrightarrow{n}_{i2C} \\ F_{i2SC} &= -[k_{neigh}(d_{i2SC} - d_{neigh}) + c_{neigh}(v_{Ri} - v_{RiSC})]\overrightarrow{n}_{i2SC} \\ F_{neigh} &= F_{i2C} + F_{i2SC} \end{aligned} \tag{3.1}$$

where k_{neigh} and c_{neigh} are the coefficients of the spring and damper. F_{i2C} and F_{i2SC} are the forces due to closest and second closest UGV respectively. d_{neigh} is the equilibrium distance between the two closest UGVs and R_i. v_{Ri} is the speed of R_i while v_{RiC} and v_{RiSC} are velocities of the closest and the second closest UGVs respectively.

Just like F_{i2C} and F_{i2SC}, the contribution by the V_L to the coordinated motion is modeled with a spring mass damper system. F_{V_L} is modeled by the sum of the spring force and damping force.

The force exerted by V_L to R_i in order to maintain a distance of d_T from T, is defined as:

$$F_{V_L} = -[k_{V_L}(d_{i2V_L} - d_T) + c_{V_L}(v_{Ri})]\overrightarrow{n}_{i2T} \tag{3.2}$$

where k_{V_L} and c_{V_L} are the constant coefficients for the spring force and damper force, respectively. d_{i2V_L} is the signed distance from R_i to V_L. d_T is the desired distance to be maintained from T to R_i when UAV is on hover over T.

The sum of the two forces, F_{V_L} and F_{neigh} gives us the dynamics of each m_{Vi} as follows:

$$m_{Vi} \begin{bmatrix} \ddot{x}_{Vi} \\ \ddot{y}_{Vi} \end{bmatrix} = F_{V_L} + F_{neigh}. \tag{3.3}$$

The reference position for the actual robot generated from the position vector of m_i. Position of the virtual reference robots can be calculated by solving the the Eqn. (3.3) and integrating twice. During this operation the velocity vector of the virtual reference robot is also generated. This velocity vector is required for calculating the reference orientation as follows:

$$\alpha_{Vi} = \arctan \frac{\dot{y}_{Vi}}{\dot{x}_{Vi}} \tag{3.4}$$

where α_{Vi} is the orientation for the virtual reference robot.

3.3 Adaptable Model Parameters

In order to successfully complete a given task with a group of robots, forming several formations might be required. The required formations are highly dependent to the given task set. Even for simple cases such goals are not trivial to accomplish.

In this work in order to achieve certain formations adaptive spring and dampers are employed. By doing so the overall freedom of the system is increased and coordinated motion is achieved in a more robust manner. The adaptable parameters are selected as smoothly varying with respect to the distance from V_L. Also the equilibrium distance of the spring is altered in order to surround the T when UAV starts hover on top of it. The tasks of the UGVs can be divided into two phase as follows:

- (A) Approaching V_L starting from an initial setting.
- (B) Achieving a circular formation with radius d_T around T, where all UGVs face towards the center.

During Phase (A), each R_i is approaching to V_L from its initial position. UGVs are away from V_L but the force, F_{neigh} is higher than F_{V_L}, because k_{neigh} is set to k_{far}, which is higher than the value of k_{V_L}. As a result, F_{neigh} dominates the equation and UGVs tend to move as a group towards V_L. Even though F_{neigh} is dominant the effect of F_{V_L} is not small, this help leading the motion of the group towards V_L in a coordinated manner. The d_{neigh} in (3.1) initialized with the value d_{far} where k_{neigh} is set to the value k_{far}. Also during this phase the value of c_{neigh} is greater than zero, by making the overall system more stiff the virtual reference robots are enforced to move together. This phase ends when d_{i2V_L} is lower than a predefined constant d_{relax} and Phase B begins.

In the second phase the UGVs goal changes to surround T. Since the UAV is able to move faster than UGVs, it already hovers over T. UGVs are able to park when they meet the specified requirements.The constants of

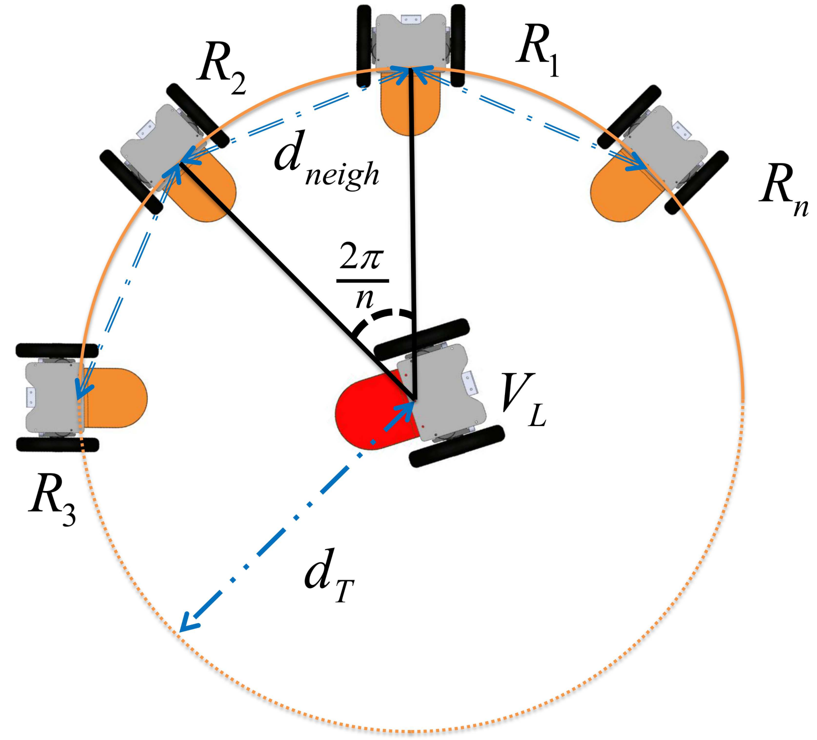


Figure 3.3: Uniform distribution of masses on the formation circle around T

coordination force are decreased, k_{V_L} is changes to a new value, k_{near}, and becomes larger than k_{neigh} and dominates the behaviour of UGVs. Also c_{neigh} is reduced to zero so the system becomes less stiff and UGV can move more freely on their own. In order to achieve an equal distribution around T the value of d_{neigh} needs to be changed to another predefined value, d_{near}, from the previous value d_{far}. In both phases while the dominant force leads, the recessive coefficient continues to affect the virtual reference robots.

The value of $dnear$ can be calculated by simply applying a bisection on the triangle in Fig. 3.3 and then applying the Law of Sines. The desired distance between UGVs $dnear$ is calculated as follows:

$$\begin{gathered}\frac{d_{near}}{2sin\left(\dfrac{\pi}{n}\right)} = \frac{d_T}{2sin\left(\dfrac{\pi}{2}\right)} \\ d_{near} = 2d_T sin\left(\dfrac{\pi}{n}\right)\ .\end{gathered} \tag{3.5}$$

In this phase instead of only moving towards T UGVs needs to start to surrounding T. Also they need to maintain a equally distributed and be d_T away from T. To do that k_{neigh} is reduced to some finite value, k_{near}, if not each R_i tends to stop when they a close enough to T, also for the same reason F_{V_L} becomes dominant.

To avoid abrupt changes in both F_{neigh} and the reference values k_{neigh} has to be reduced in a smooth continuous manner. Because of this reason k_{neigh} is designed as a function of the distance of m_{Vi} to V_L. By doing so the jumps in the reference values are avoided and the stability of the system is preserved. k_{neigh} is designed as continuous function of d_{i2T} by employing sigmoid functions as follows:

$$k_{neigh} = k_{near} + \frac{k_{far} - k_{near}}{1 + \exp(\eta(d_{relax} - d_{i2T} + \zeta))}\ , \tag{3.6}$$

where η, ζ are positive constants that defines the characteristics of the sigmoid curve. d_{i2V_L} is the signed distance between m_{Vi} and V_L. k_{far} and k_{near} are the predefined values, the desired values when UGVs far from and near to the V_L respectively. In Fig. 3.4 the obtained continuous switching function k_{neigh} is depicted.

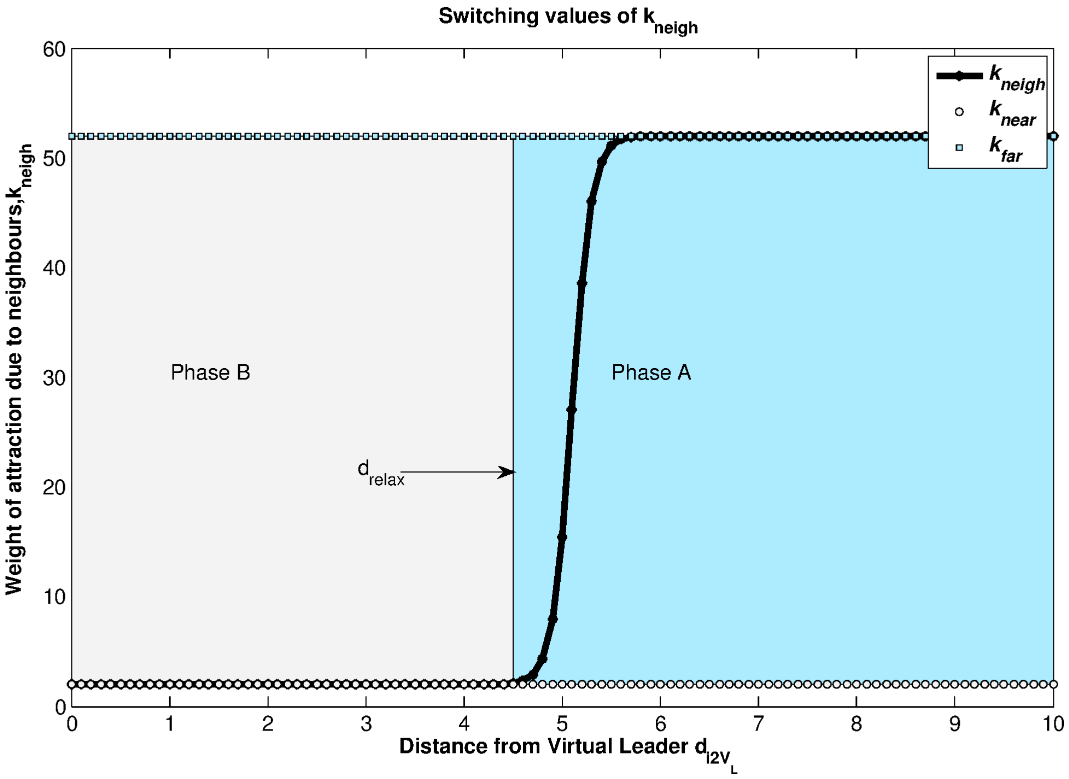


Figure 3.4: Adaptive spring coefficient, k_{neigh} vs distance from T, d_{i2V_L}

3.4 Switching Between Controllers for UGVs

The details of the control laws for both trajectory tracking and for parking is given in Chapter 2. By employing the reference trajectories generated by virtual masses UGVs tries minimize the distance between themselves and the V_L by using the control law (2.11). The position of V_L is updated several times, till UAV starts to hover on the actual target, T. UAV continues to move till it reaches on top of T, when UAV starts hover on top of T another signal is also broadcast. This signal allows the UGVs to initiate the park mode when they are close enough to radius, d_T, around T. After that the goal becomes uniformly surrounding around the radius of the circle. Finally, the control law in Eqn.(2.14) is used in order to park the UGVs headed towards T with fixed position reference. UAV hovers on target till UGV successfully surrounds T.

3.5 Simple Path Planning for UAV

The trajectory of the UAV is very important for successful coordination of the UGVs. The UAV directly effect the trajectories of UGV by the contribution of $\overrightarrow{v}_{i2V_L}$ on $\overrightarrow{v}_{des}$. Similar to UGVs the UAV has also two different phase but these phase does not require change in the controller or the adaptive variables since this problem can be tackled as trajectory tracking problem of UAV which was described in Chapter 2.

The trajectory of UAV is tackled flying from an initial point to a final point without any stopping. The initial point is a close point to UGVs due to the defined previously scenario. The final point is on top of T. The UAV is able to locate the position of the T by facilitating a visual sensor. The trajectory from initial to final position is calculated by employing quintic order polynomials and given as reference to the position controller of the UAV.

Chapter 4

A Kinematic Coordination Model

In this chapter the details of the second coordination algorithm between UAV and UGVs is described. The coordination of UGVs is based on a kinematic reference generation model [54], and the coordination is achieved by properly designing the linear and angular velocities of each UGV. The reference position and orientation for each UGV, R_i, is generated by a virtual reference robot. Since the reference velocities are generated in a manner that abides the nonholonomic constraint, actual robots are also able to track the generated trajectory. The generated velocities are integrated to compute reference position and orientation for each R_i. The regulation of the position and orientation errors between the virtual reference robot's pose and the actual robot's pose to zero are guaranteed with a smooth time-varying feedback control law as described Chapter 2. The projection of the UAV's position onto the X-Y plane is defined to be the position of the virtual leader V_L. Coordination of UGVs is achieved by defining desired velocities with respect to closest neighbors of R_i and V_L with smoothly varying gains. Also the trajectory reference used for the UAV is described in detail.

In Fig. 4.1 a scenario is depicted with four UGV using kinematic coordination model. As it can be seen in the figure the fourth UGV is not related with the course of R_i. The relation between R_i with its two closest neighbour and V_L is also depicted.

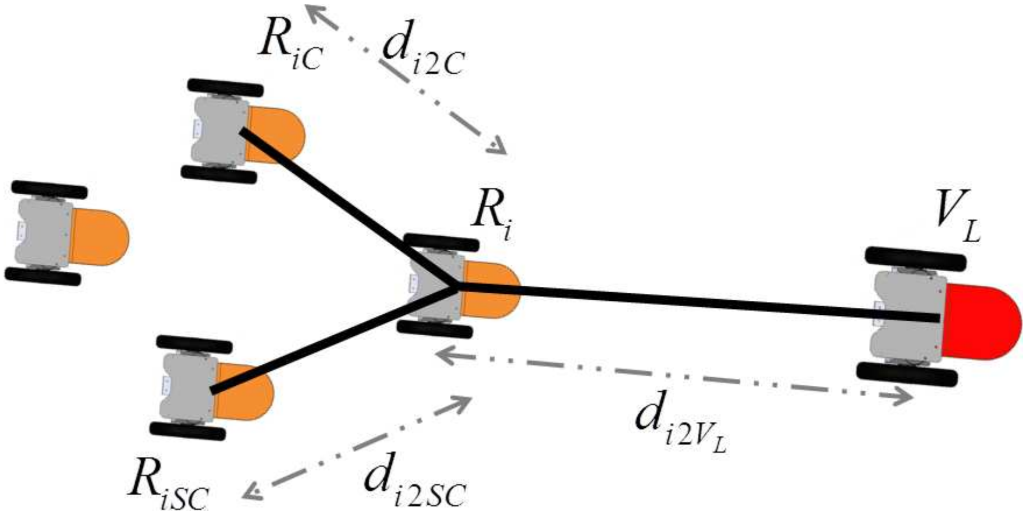


Figure 4.1: The relation between R_i with its two closest neighbors and V_L.

4.1 Desired Velocities of UGVs

Coordinated motion of UGVs is achieved by defining desired velocities with respect to closest neighbors of R_i and V_L. The details for desired velocity due to virtual leader and due to the neighbours are explained in this section.

4.1.1 Desired Velocity due to Virtual Leader

In order to accomplish a coordinated task each R_i is moving towards V_L from arbitrary initial pose. Each R_i has a desired velocity vector towards V_L, $\overrightarrow{v}_{i2V_L}$ from its current position. The $\overrightarrow{v}_{i2V_L}$ is defined to move R_i towards V_L until UAV reaches on top of T, then each R_i maintains a given distance d_T between itself and T, as follows:

$$\overrightarrow{v}_{i2V_L} = k_{lin}(d_{i2V_L} - d_T)\overrightarrow{n}_{i2V_L} \ , \tag{4.1}$$

where $k_{lin} > 0$ is a constant, $\overrightarrow{n}_{i2V_L}$ is the unit vector from R_i to V_L, and d_{i2V_L} is the distance from R_i to V_L. As it can be seen in Eqn.(4.1) the effect of this velocity is high when the UGVs are far away from V_L.

4.1.2 Desired Velocity due to Neighbors

In many coordination scheme the agents require to interact with each other. The proposed algorithm employ an interaction between R_i and its two nearest neighbors. Each R_i is required to maintain a distance, d_{neigh} to its closest neighbors in order to move in a coordinated manner. The second closest neighbour loses its effect on the desired velocity after reaching a predefined distance, d_{relax}, from V_L. The position of V_L is updated several times per second, till UAV starts to hover on the actual target, T.

The effect of the closest neighbours is included in order to properly design the desired velocities for each R_i. The coordinated motion can be described as maintaining a certain distance while moving together in a synchronous manner. In that aspect maintaining a certain distance between the R_i is crucial. In other words, each R_i is required to maintain a certain distance

d_{neigh} from its closest neighbour,R_{iC}, second closest neighbour, R_{iSC} in order to accomplish coordinated motion. The desired velocity vector of R_i due to its closest neighbors, $\overrightarrow{v}_{neigh}$, is given by:

$$\begin{gathered}
\overrightarrow{v}_{neigh} = \overrightarrow{v}_C + \overrightarrow{v}_{SC} \ , \\
\overrightarrow{v}_C = k_{lin}(d_{i2C} - d_{neigh})\overrightarrow{n}_{i2C} \ , \\
\overrightarrow{v}_{SC} = \begin{cases} k_{lin}(d_{i2SC} - d_{neigh})\overrightarrow{n}_{i2SC} & \text{if} \quad d_{i2V_L} \geq d_{relax} \\ 0 & \text{if} \quad d_{i2V_L} < d_{relax} \end{cases} \ ,
\end{gathered} \tag{4.2}$$

where d_{i2C} and d_{i2SC} are the distances from the closest neighbours.$\overrightarrow{n}_{i2C}$ and $\overrightarrow{n}_{i2SC}$ are unit vectors, from each R_i to its two closest neighbors. $\overrightarrow{v}_C$ and $\overrightarrow{v}_{SC}$ are velocities due to two closest neighbors respectively. d_{i2V_L} is the distance of each R_i to V_L and k_{lin} is a positive constant. d_{neigh} is the coordination distance R_i should maintain between itself and its closest neighbors, and d_{relax} is the critical distance of R_i to V_L below which the second closest neighbor loses effect.

4.1.3 Combination of Desired Velocities for Reference Generation

The desired velocity vector, $\overrightarrow{v}_{des}$, of R_i is defined as a convex combination of previously calculated velocities $\overrightarrow{v}_{i2V_L}$ and $\overrightarrow{v}_{neigh}$ in Eqn.(4.1) and in Egn. 4.2 respectively. $\overrightarrow{v}_{des}$ is calculated as:

$$\overrightarrow{v}_{des} = k_{V_L}\overrightarrow{v}_{i2V_L} + k_{neigh}\overrightarrow{v}_{neigh} \ , \tag{4.3}$$

where k_{V_L} is the coefficient of velocity due to V_L, and k_{neigh} is the coefficient of the velocity due to the closest neighbors of R_i. These two coefficients defines the dependence of the generated reference velocity, on the neighbors and V_L till the coordinated motion is successfully accomplished. Since these coefficients have a major role on the coordination algorithm it is beneficial to define them as adaptive parameters in order to add extra degree of freedom to the system. The desired velocity combination is integrated in order to calculate the reference trajectories for each R_i.

4.2 Parameter Switching for UGVs

Successfully completing a given goal with a robot group in a coordinated manner is a complicated task. In order to accomplish a given task with a group of robots, forming several formations might be required. Also the requirements for a task may change with respect to each scenario and to the described goals.

In this section to achieve coordinated motion in a robust manner, the gains of the velocity coefficients are selected as smoothly varying with respect to the distance from V_L. By doing so the overall freedom of the system is increased and coordinated motion can be achieved in a robust manner. Also the coordination distance of the UGVs is altered in order to surround the T when UAV starts hover on top of it. The tasks of the UGVs can be divided into two phase as follows:

- (A) Approaching V_L starting from an initial setting.
- (B) Achieving a circular formation with radius d_T around T, where all UGVs face towards the center.

During Phase (A), each R_i is approaching to V_L from its initial position. UGVs are away from V_L but the weight of coordination coefficient,k_{neigh} is higher than k_{V_L}. As a consequence UGVs tend to move as a group towards V_L. The d_{neigh} in (4.2) initialized with the value d_{far} where k_{neigh} is set to the value k_{far}. This phase ends when d_{i2V_L} is lower than a predefined constant d_{relax} and Phase B begins. During the second phase the UGVs goal changes to surround T. Since the UAV is able to move faster than UGVs it is already over T and hovers.UGVs are able to park when they meet the specified requirements. In this phase k_{V_L} is changes to a new value, k_{near}, and becomes larger than k_{neigh} and dominates the behaviour of UGVs. Instead of moving towards T, UGVs start to surround it. In order to achieve an equal distribution around T the value of d_{neigh} needs to be changed to another predefined value, d_{near}, from the previous value d_{far}. In both phases the recessive coefficient continues to affect the reference velocity values.

In order to be able to calculate the previously mentioned varying coefficients d_{neigh}, k_{neigh} and k_{V_L} are designed as continuous function of d_{i2T} by employing sigmoid functions. The function for d_{neigh}:

$$d_{neigh} = d_{near} + \frac{d_{far} - d_{near}}{1 + exp(\eta(d_{relax} - d_{i2V_L} + \zeta))} \ . \tag{4.4}$$

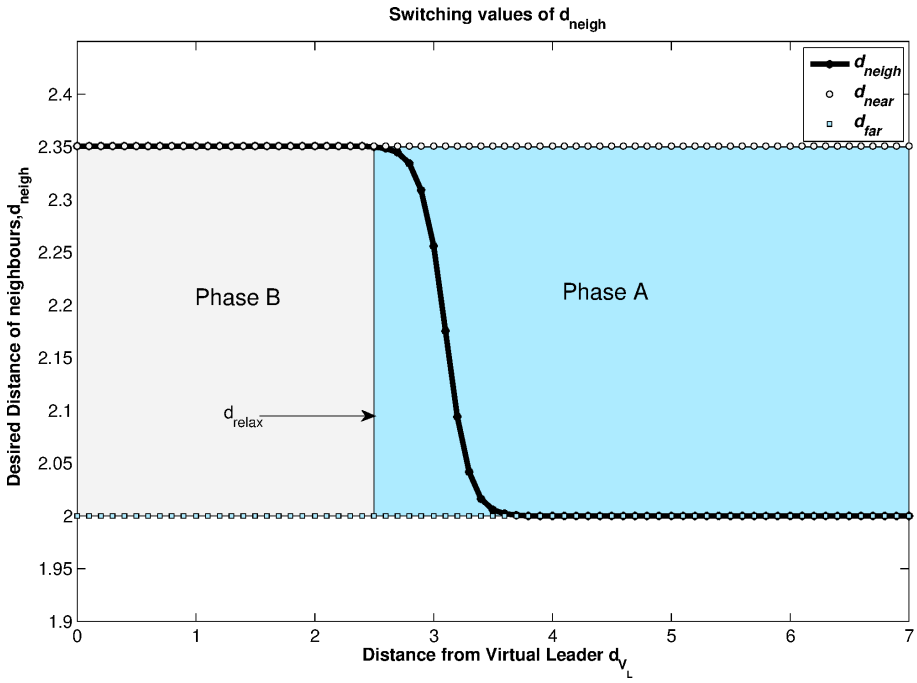


Figure 4.2: Continuously changing coordination distance, d_{neigh} with respect to distance from T, d_{i2V_L}

where η, ζ are positive constants that defines the characteristics of the sigmoid curve. d_{far} and d_{near} are the predefined distance values, the desired distances between UGVs far from and near to the T respectively. Depending on the number of robots surrounding the T, value of d_{near} is calculated as follows:

$$d_{near} = 2d_T sin\left(\frac{\pi}{n}\right) . \tag{4.5}$$

The details of calculating d_{near} using basic geometry is discussed in Chapter 3. Continuous switching behaviour of d_{neigh} with respect to the distance d_{i2V_L}

is depicted in Fig. 4.2.

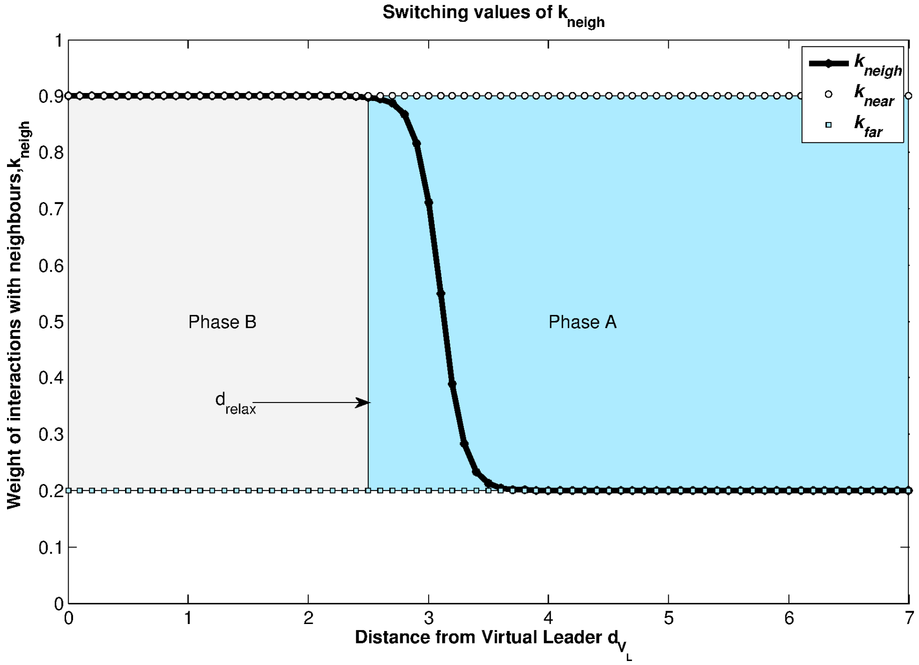


Figure 4.3: Continuously changing coordination distance, k_{heigh} with respect to distance from T, d_{i2V_L}

Similarly k_{neigh}, is designed as a continuous function with respect to the distance d_{i2V_L} ,by employing the following sigmoid function:

$$k_{neigh} = k_{near} + \frac{k_{far} - k_{near}}{1 + exp(\eta(d_{relax} - d_{i2T} + \zeta))} \ , \tag{4.6}$$

where η, ζ are positive constants that defines the characteristics of the sigmoid curve. k_{far} and k_{near} are the predefined values, the desired values when UGVs far from and near to the V_L respectively. In Fig. 4.3 the obtained continuous switching function k_{neigh} is depicted.

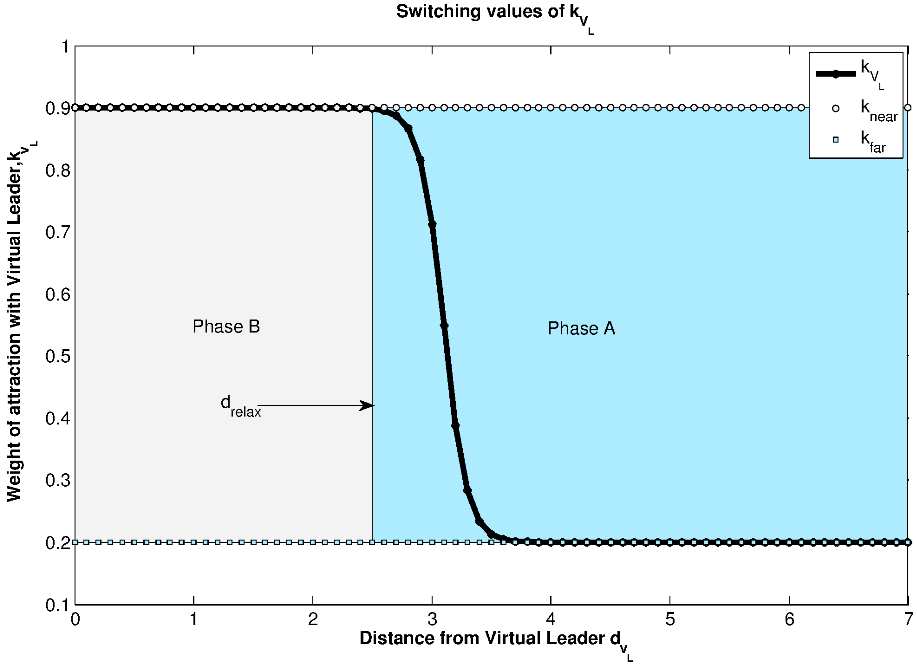


Figure 4.4: Continuously changing coordination distance, k_{V_L} with respect to distance from T, d_{i2V_L}

In order to reduce computational load k_{V_L} is designed as a function of k_{neigh}. By doing so a new continuous function is developed at the cost of a basic operation. In Fig. 4.4 the obtained continuous switching function k_{V_L} is depicted.

$$k_{V_L} = 1 - k_{neigh} \; . \tag{4.7}$$

It should be noted that the values of of k_{near} and k_{far} are selected in a manner that both k_{neigh} and k_{V_L} are greater than zero and less than one.

4.3 Reference Trajectory Generation for UGVs

The reference linear and angular velocities, $u_{1_{ref}}$ and $u_{2_{ref}}$ are calculated by using the desired velocity v_{des} given in 4.3. In order to achieve a trackable references smooth and continuous reference values are generated for the UGVs. As previously discussed the velocity reference for UGVs can be expressed in terms of the linear and angular velocities because of the nonholonomic constraints. α_{des} and v_{des}, the reference orientation and reference speed respectively, is calculated as given below:

$$\begin{aligned} \alpha_{des} &= \arctan(v_{y_{des}}/v_{x_{des}}) \\ |v_{des}| &= \sqrt{v_{x_{des}}^2 + v_{y_{des}}^2} \ , \end{aligned} \tag{4.8}$$

For each R_i , the angular velocity reference is designed in terms of the the orientation error, e_α, defined by the following equation as:

$$\begin{aligned} e_\alpha &= \alpha - \alpha_{des} \ , \\ u_{2_{ref}} &= -k_{rot} e_\alpha \ . \end{aligned} \tag{4.9}$$

where k_{rot} is a proportional gain greater than zero. The linear velocity reference of each R_i, $u_{1_{ref}}$, is designed with a piecewise linear function in order to enable saturation of the linear velocity. $u_{1_{ref}}$ can be calculated as follows:

$$u_{1_{ref}} = \begin{cases} (1/\left|e_\alpha\right|)\left|v_{des}\right|, & \text{if} \quad \left|e_\alpha\right| \geq \alpha_{lim} \\ (1/\alpha_{lim})\left|v_{des}\right|, & \text{if} \quad \left|e_\alpha\right| < \alpha_{lim} \end{cases} \ , \tag{4.10}$$

where α_{lim}, a saturation value to limit the linear velocity, is greater than zero.

The reference pose of R_i, can be obtained by the integration of the calculated reference linear and angular velocities:

$$\begin{bmatrix} x_{ref} \\ y_{ref} \\ \alpha_{ref} \end{bmatrix} = \begin{bmatrix} \int u_{1ref} cos\alpha_{ref} dt \\ \int u_{1ref} sin\alpha_{ref} dt \\ \int u_{2ref} dt \end{bmatrix}. \tag{4.11}$$

4.4 Switching Between Controllers for UGVs

The control laws that governs the trajectory tracking and parking of the UGV is given in Chapter 2. The reference trajectories are generated by the integrating reference given by the control law given in Eqn.(2.11). The UGVs tries minimize the distance between themselves and the V_L. The position of V_L is updated several times per second, till UAV starts to hover on the actual target, T. UAV continues to move till it reaches on top of T, when UAV starts hover on top of T another signal is also broadcast. This signal allows the UGVs to initiate the park mode when they reach a predefined radius, d_T, around T. After that the goal becomes uniformly surrounding around the radius of the circle. Finally, the control law in Eqn.(2.14) is used in order to park the UGVs headed towards T with fixed position reference. UAV hovers on target till UGV successfully surrounds T.

4.5 Path Planning for UAV

The trajectory of the UAV greatly affects the coordinated motion of the UGVs. This is due to the effect of $\overrightarrow{v}_{i2V_L}$ on $\overrightarrow{v}_{des}$. Similar to UGVs the UAV has also two different phase, following a trajectory and hover. These two phases does not require change in the controller or the adaptive variables used for UGVs coordination. This problem can be tackled as a trajectory tracking problem of UAV which was described in Chapter 2.

The trajectory of UAV is generated that UAV fly from an initial point to a final point without any stop. The initial point is a close point to UGV group due to the previously defined scenario. The final point is on top of T. The UAV is able to locate the position of the T by utilizing a visual sensor. The trajectory from initial to final position is calculated by employing quintic order polynomials and given as reference to the position controller of the UAV.

The trajectory of the UAV needs to became elongated and more elaborate because of the obstacles introduced in this Chapter. In order to do so a via-point based approach introduced. The obstacles are assumed to be detected and mapped with a visual sensor by UAV. In this thesis the reference trajectory for UAV is generated by using probabilistic method in the environment with obstacles.

To generate reference trajectory for UAV a map based solution is preferred, basically to use the advantages of an aerial vehicle. The UAV can easily take pictures and come up with a possible route for itself or for UGVs by detecting important features. In literature there are several amp based path planning algorithms such as; Potential Fields,Voronoi Roadmap Method, D*

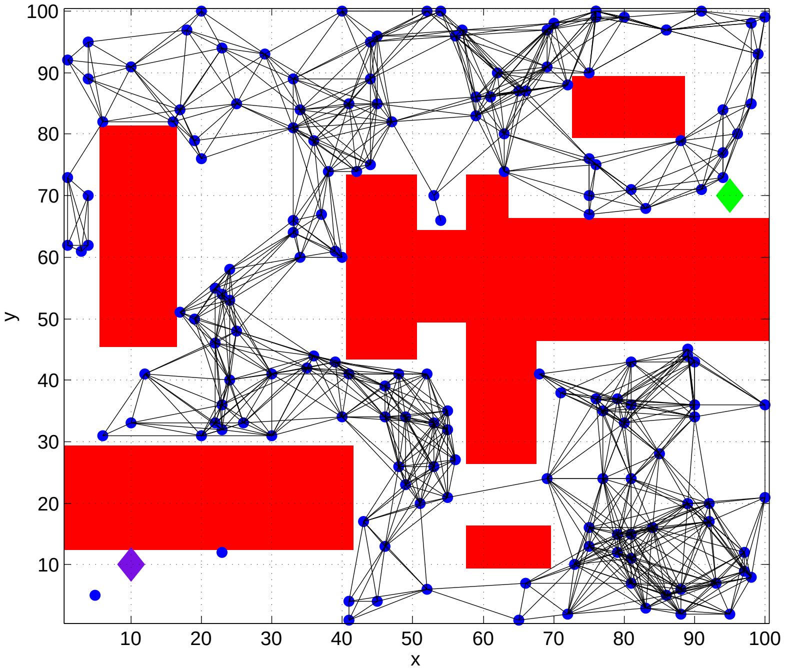


Figure 4.5: An unsuccessful path planning example with Probabilistic Roadmap Method. Purple diamond is start point and green diamond represents goal position.

algorithm, Distance Transforms, Cell Decomposition Methods, Probabilistic Roadmap Method and Rapidly-exploring Random Tree Method. All these methods are well studied and known for their advantage an disadvantages and details of these algorithm can be found in [40, 55]

In order to generate the path Probabilistic Roadmap Method is used, favoring its feasibility even in large scale maps. The input to that algorithm is a map and n, the number of random points are distributed over the free space of map. The result of this algorithm is a set of points, containing both x and y coordinate information for each point, from an initial position to a goal position with arbitrary number of elements depending on both n and the obstacles in the map.

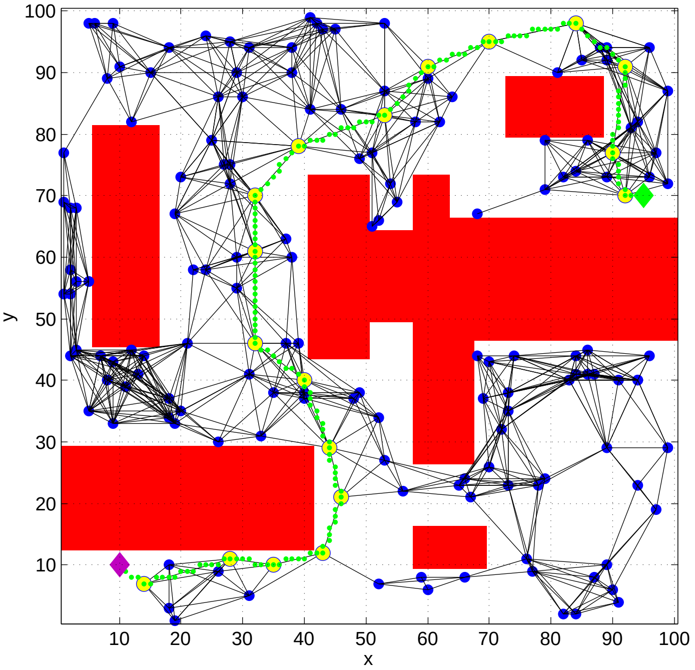


Figure 4.6: An successful path planning example with Probabilistic Roadmap Method. Purple diamond is start point, green diamond represents goal position. Yellow circles are the via points for the trajectory of UAV.

In Fig.(4.5) and Fig.(4.6) two possible outcome of the Probabilistic Roadmap Method is shown. In both figures purple diamond is start point, green diamond represents goal position. Blue circles represents the random points in the map. As it can be seen in Fig.(4.5) the points are not equally distributed, there is a high concentration of random points at the down right corner of the map while there are very few around the initial point. Due to that there is no possible solution for this case. On the other hand in Fig.(4.5) it can be seen that there is a possible path consisting of twenty points, including the initial and final positions.

These points are used as via points for the reference of UAV. A position trajectory is generated using quintic trajectories by connecting two consecutive via points one after another till the final point is reached. The position reference is used to generate the orientation references as described in Chapter 2.

Chapter 5

Simulation Results

In this chapter both of the proposed methods are verified by simulations and animations in MATLAB/Simulink. The dynamic coordination model is verified with three and four robots on simple trajectory. The kinematic model is verified with three and five robots on simple trajectories.

Due to its advantages on the generated references the kinematic reference model provided more robust results. In order to test the validity of the prosed algorithm, a more complicated trajectory for UAV is given as a multiple via point trajectory while UGVs are coordinated by kinematic coordination model. Simulations were run with three and five robots. The results showed the feasibility of the algorithm on complex trajectories.

5.1 Simulations with Dynamic Coordination Model

To verify our first proposed method we have used both simulations and animations in MATLAB/Simulink. In simulations maximum angular speed for each UGV was set to $\frac{\pi}{3}$ rad/s while maximum linear speed was 1 m/s. UGVs are assumed to be motionless and UAV is assumed to be in hover at the beginning of simulations. UAV broadcasts its position information to the UGVs at 3 Hz and a 100 ms communication delay is assumed. We also added 100 ms communication delay between the UGVs. We have simulations for groups of three and four UGVs to collaborate with UAV in order to find and surround the T. The values used for in Dynamic Coordination Model are tabulated in Table 5.1.

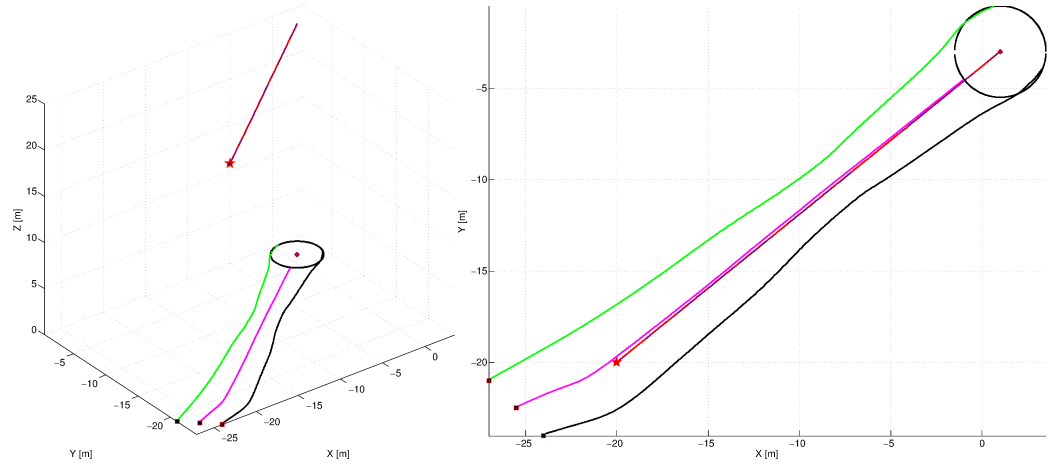
Table 5.1: Dynamic Coordination Parameters

Coordination Variable Name	Value
k_{near}	2
k_{far}	52
d_{relax}	3
d_{far}	2.45
d_{near}	2
η	10
ζ	0.5

5.1.1 UAV and 3 UGVs

In the initial setting, the UGVs are placed at the corner of a rectangular area while T is at the center. It was observed that they approach each other and move towards T in a coordinated manner, under the guidance of UAV.

The UAV waits over T while broadcasting the position of the V_L. After that, when the UGVs are close enough to T they start to spread, in order to achieve neighboring distances of d_{near}. The 3-D trajectories for UGVs and UAV can be seen in the Figure 5.1(a). The trajectories of UGVs and UAV are depicted on the horizontal plane in Figure 5.1(b). As it can be seen from Figure 5.2(d) the group achieves the desired formation in the final configuration. Also the zoomed version of Figure 5.2 can be seen in Figure 5.3. The flight information with given reference values of the quadrotor can be seen in Figure 5.4 and Figure 5.5, respectively.



(a) **(b)**

Figure 5.1: Simple Dynamical Model with Three Robots: Trajectories of UAV and three UGVs; (a) in 3D view, (b) in 2-D view, on X-Y plane.

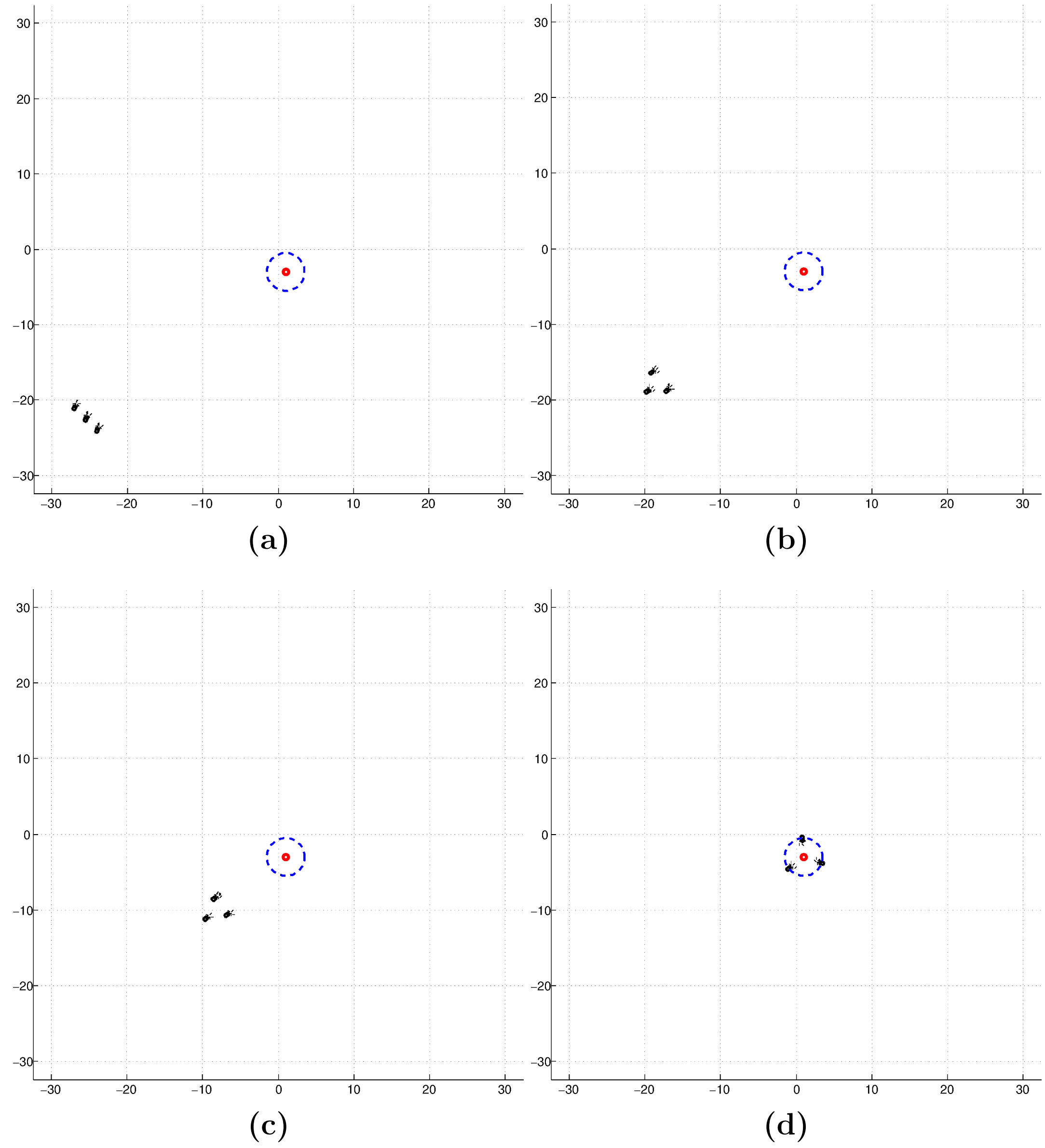


Figure 5.2: Simple Dynamical Model with Three Robots: (a) Initial configuration, (b) Coordinated motion, (c)Starting to surround T, (d) Desired formation.

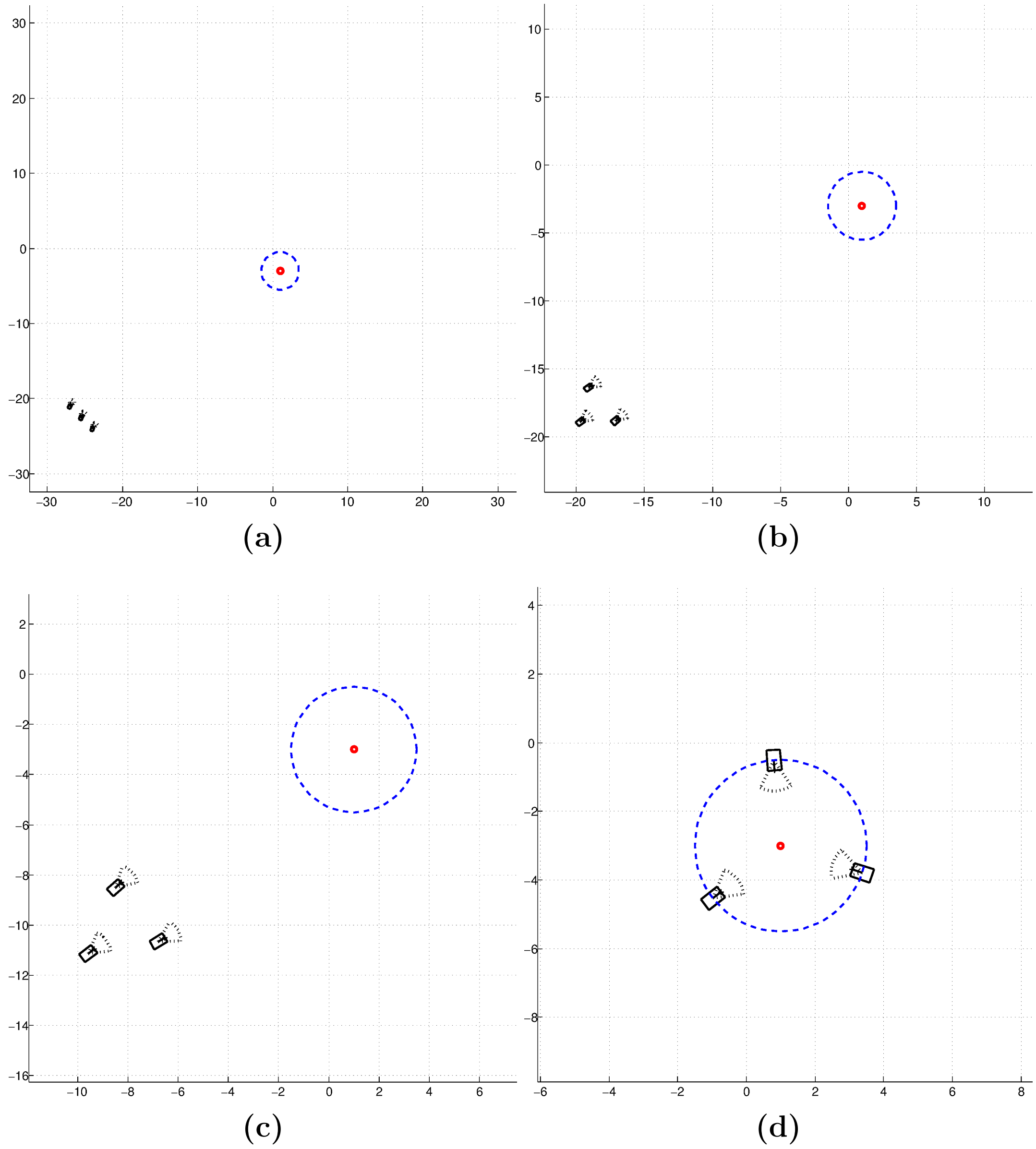


Figure 5.3: Simple Dynamical Model with Three Robots :(a) Initial configuration, (b) Zoomed version of coordinated motion, (c) Zoomed version of starting to surround T, (d) Zoomed version of desired formation.

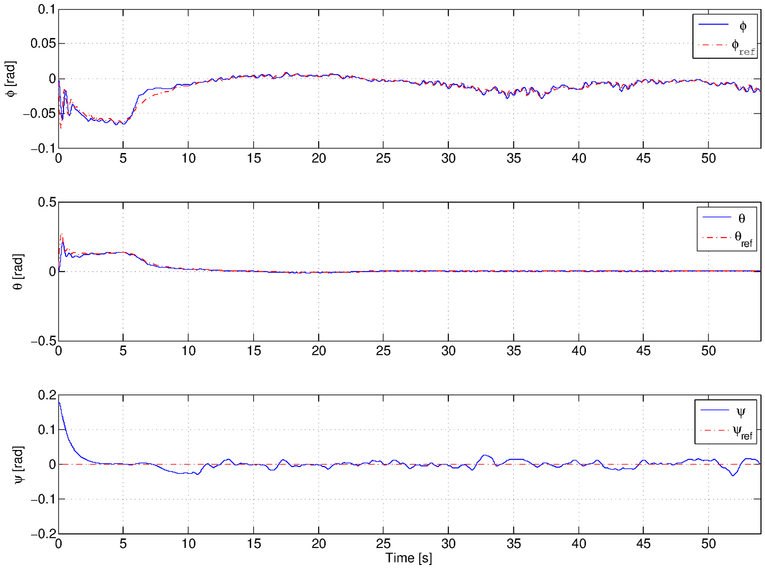


Figure 5.4: ϕ, θ, ψ of UAV with calculated reference values.

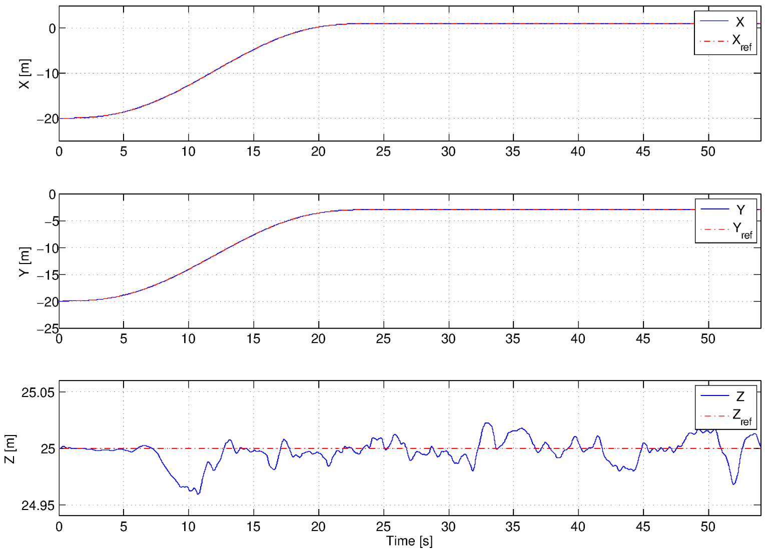
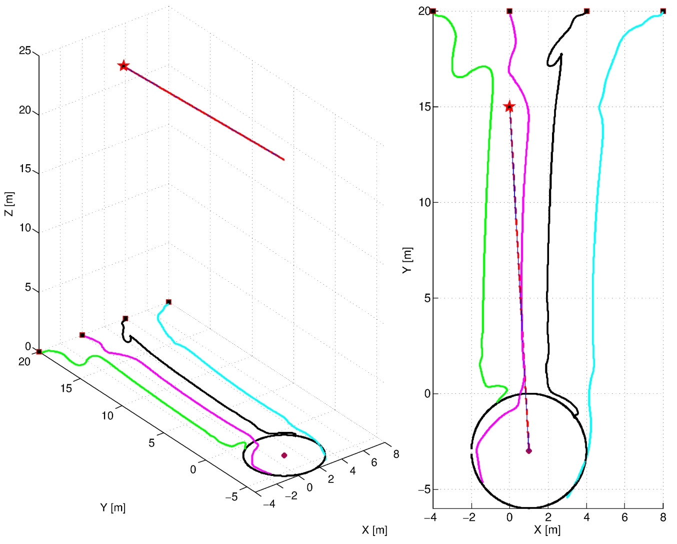


Figure 5.5: X, Y, Z of UAV with reference values.

5.1.2 UAV and 4 UGVs

The 3-D trajectories for UGVs and UAV can be seen in the Figure 5.6(a). The trajectories of UGVs and UAV are depicted on the horizontal plane in Figure 5.6(b). In this simulation the UGVs are placed as a row to the upper side of a rectangular area while T is at the lower side.The UAV is located slightly ahead of UGVs. The initial setting of UGVs can be seen in Figure 5.7(a). Since there are four UGVs, the risk of collision increases. In Figure 5.6(b) it can be seen that there is backward motion of a UGV, the one shown with black, in order to avoid a collision. After UAV reaches on top of the T, the UGVs received a signal that allowed them to park. Since UAV is faster than the UGVs, UAV hovers on. The final formation is successfully achieved with a uniform distribution of UGVs as shown in Figure 5.7(d).



(a) (b)

Figure 5.6: Simple Dynamical Model with Four Robots: Trajectories of UAV and four UGVs; (a) in 3D view, (b) in 2-D view, on X-Y plane.

The zoomed version of Figure 5.7 can be seen in Figure 5.8. Also the flight information and reference values of the quadrotor can be seen in Figure 5.9 and Figure 5.10, orientation and position, respectively.

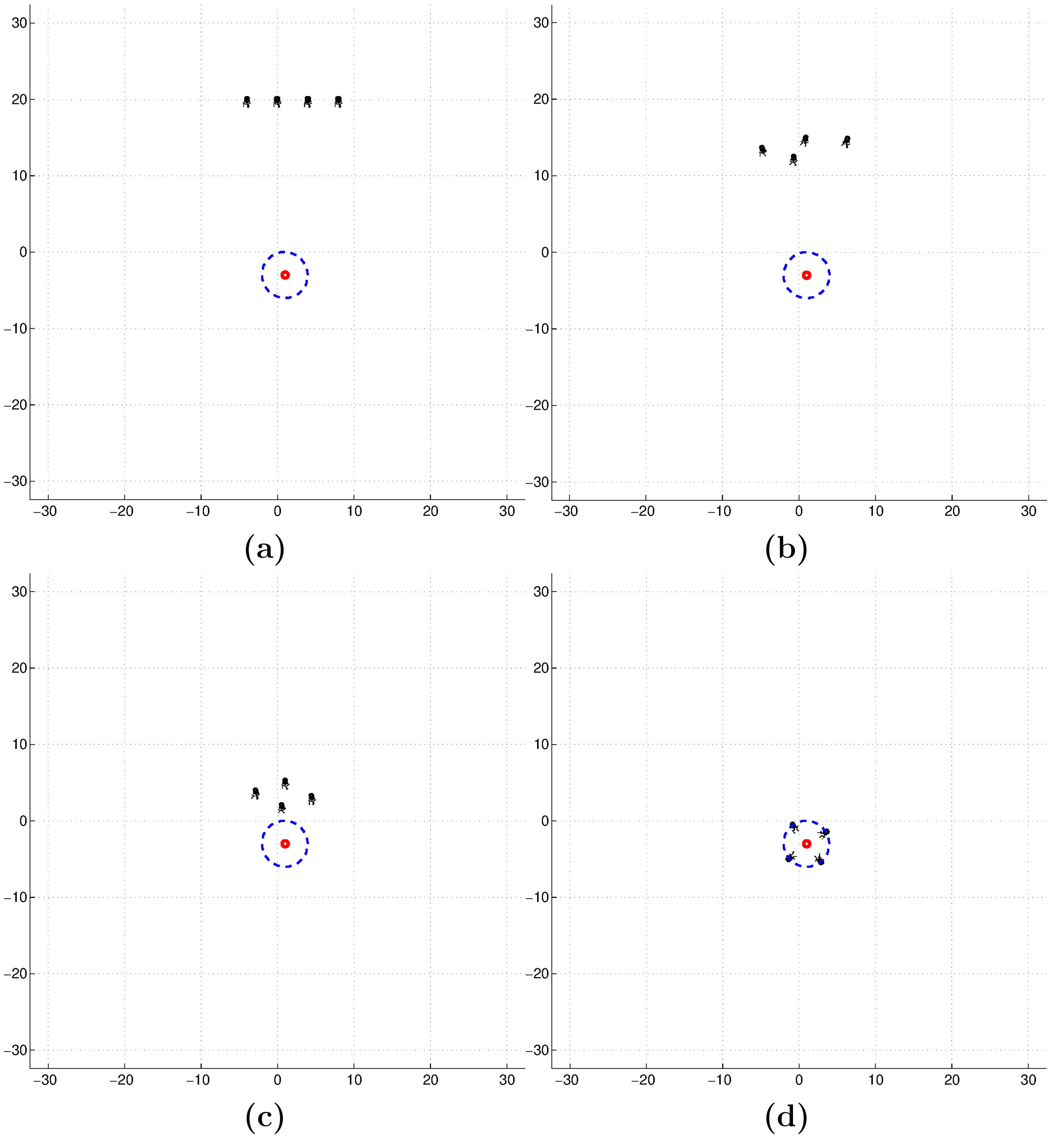


Figure 5.7: Simple Dynamical Model with Four Robots: (a) Initial configuration, (b) Coordinated motion, (c) Circular motion around target, (d) Desired formation.

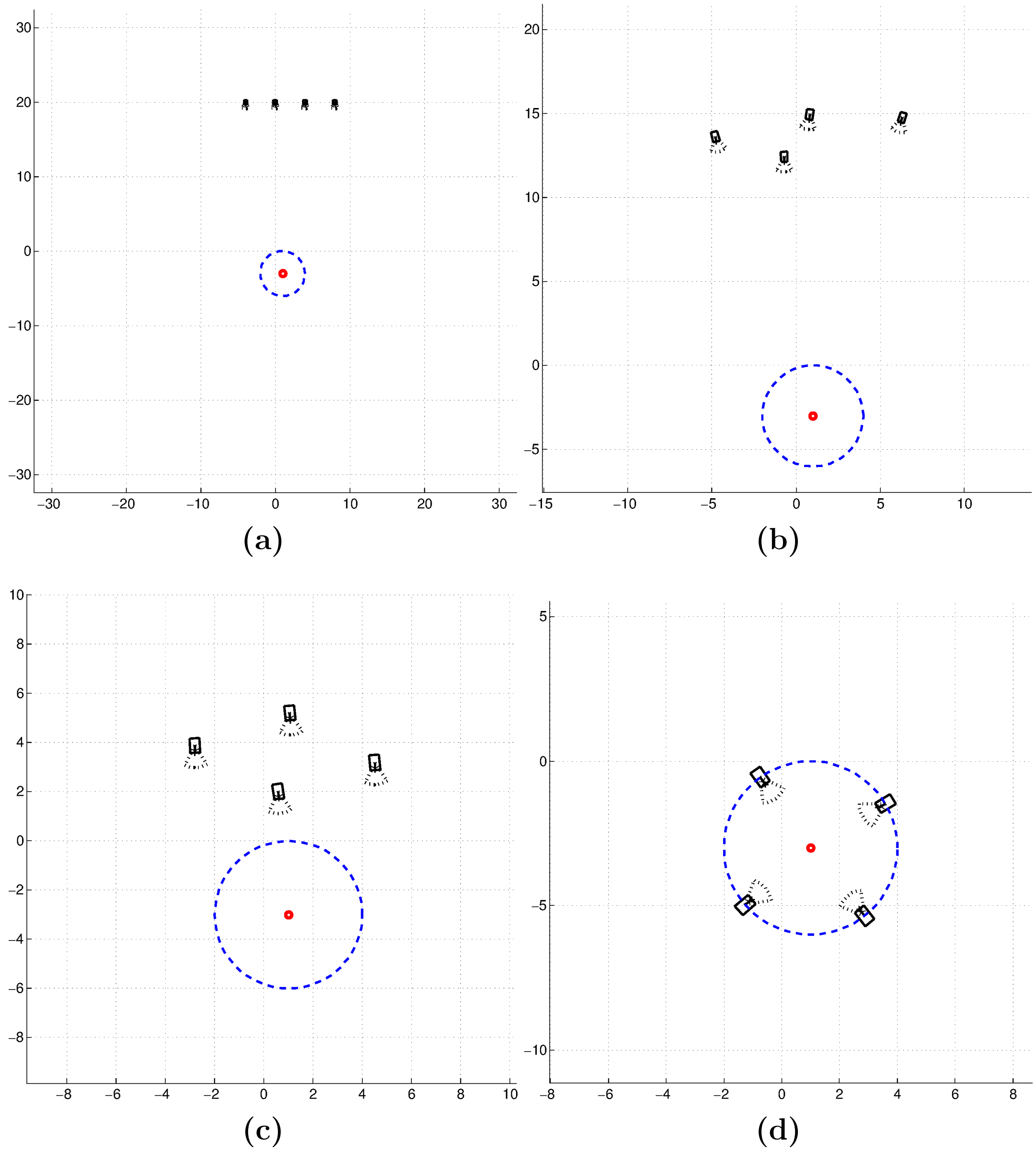


Figure 5.8: Simple Dynamical Model with Four Robots: (a) Initial configuration, (b) Zoomed version of coordinated motion, (c) Zoomed version of starting to surround T, (d) Zoomed version of desired formation.

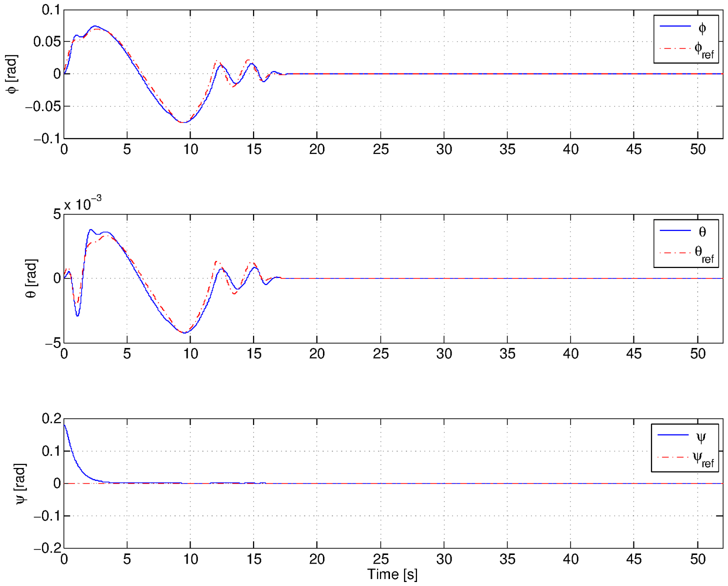


Figure 5.9: ϕ, θ, ψ of UAV with calculated reference values.

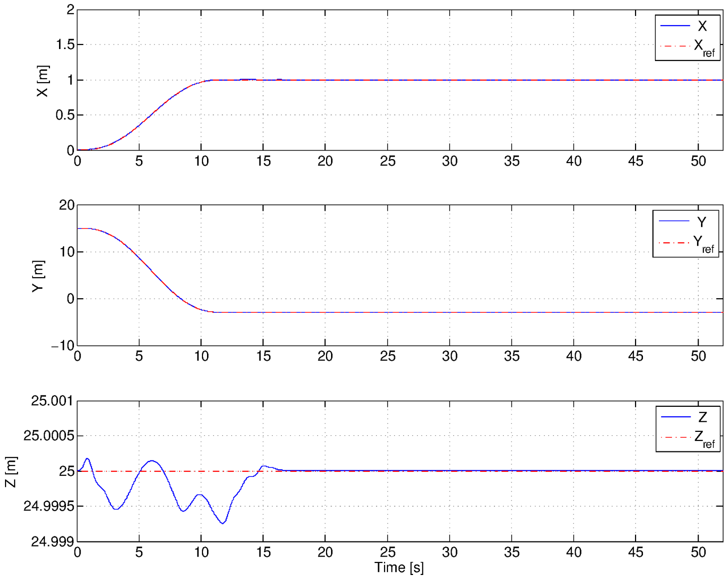


Figure 5.10: X, Y, Z of UAV with reference values.

5.2 Simulations for Kinematic Coordination Model

To verify our second proposed coordination model for simple and via-point trajectories we have used both simulations and animations in MATLAB/Simulink. In these simulations maximum angular speed for each UGV was set to $\frac{\pi}{3}$ rad/s while maximum linear speed was 1 m/s. UGVs are assumed to be motionless and UAV is assumed to be in hover at the beginning of these simulations. UAV broadcasts its position information to the UGVs at 1 Hz, and a 500 ms communication delay is assumed. We also added 100 ms communication delay between the UGVs.The values used for in Kinematic Coordination Model are tabulated in Table 5.2. We have simulations for groups of three and five UGVs to collaborate with UAV in order to find and surround the T for both trajectory types.

Table 5.2: Kinematic Coordination Parameters

Coordination Variable Name	Value
k_{near}	0.8
k_{far}	0.1
d_{relax}	5
d_{far}	2.35
d_{near}	2
α_{lim}	1
η	10
ζ	0.5

5.2.1 Simulations with simple trajectory

UAV and 3 UGVs

Similar to the initial setting in 5.1.1, the UGVs are placed at the corner, but this time upper corner, of a rectangular area while T is close to center. It was observed that they approach each other and move towards T in a coordinated manner, under the guidance of UAV. The UAV waits over T while broadcasting the position of the V_L. After that, when the UGVs are close enough to T they start to spread, in order to achieve neighboring distances of d_{near}. During this spreading motion each R_i prevents collision while they are still trying to stay on the circle due to $\overrightarrow{v}_{i2V_L}$; hence perform circular motion. The 3-D trajectories for UGVs and UAV can be seen in the Figure 5.11. The trajectories of UGVs and UAV are depicted on the horizontal plane in Figure 5.12.

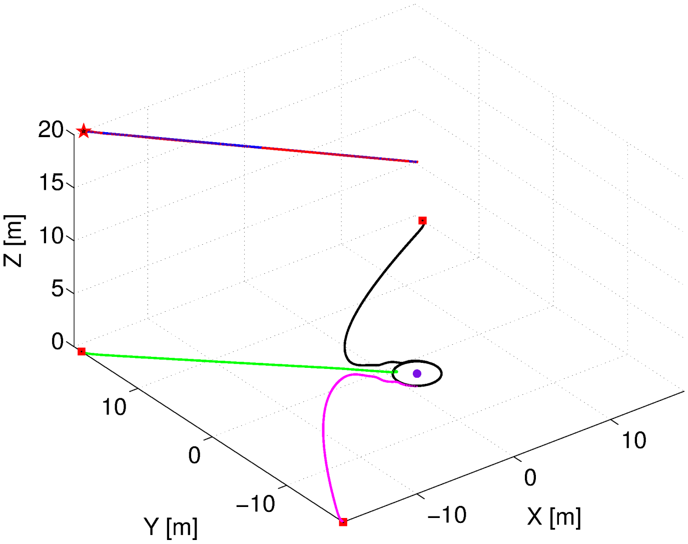


Figure 5.11: Kinematic Model with Three Robots: Trajectories of UAV and three UGVs in 3D view.

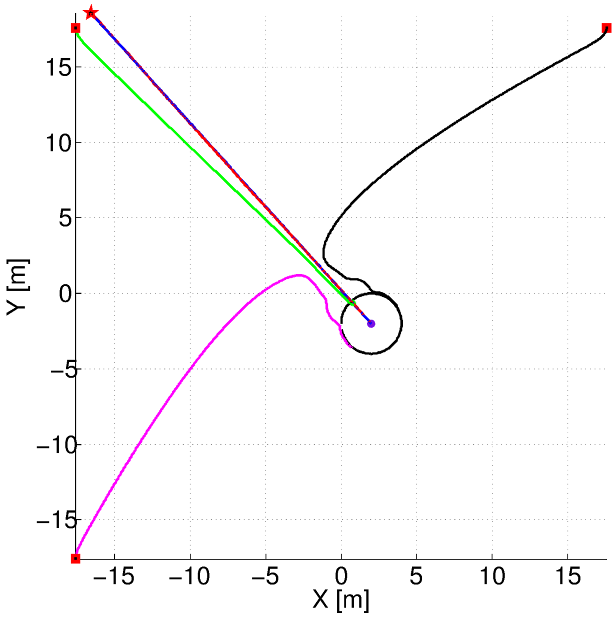
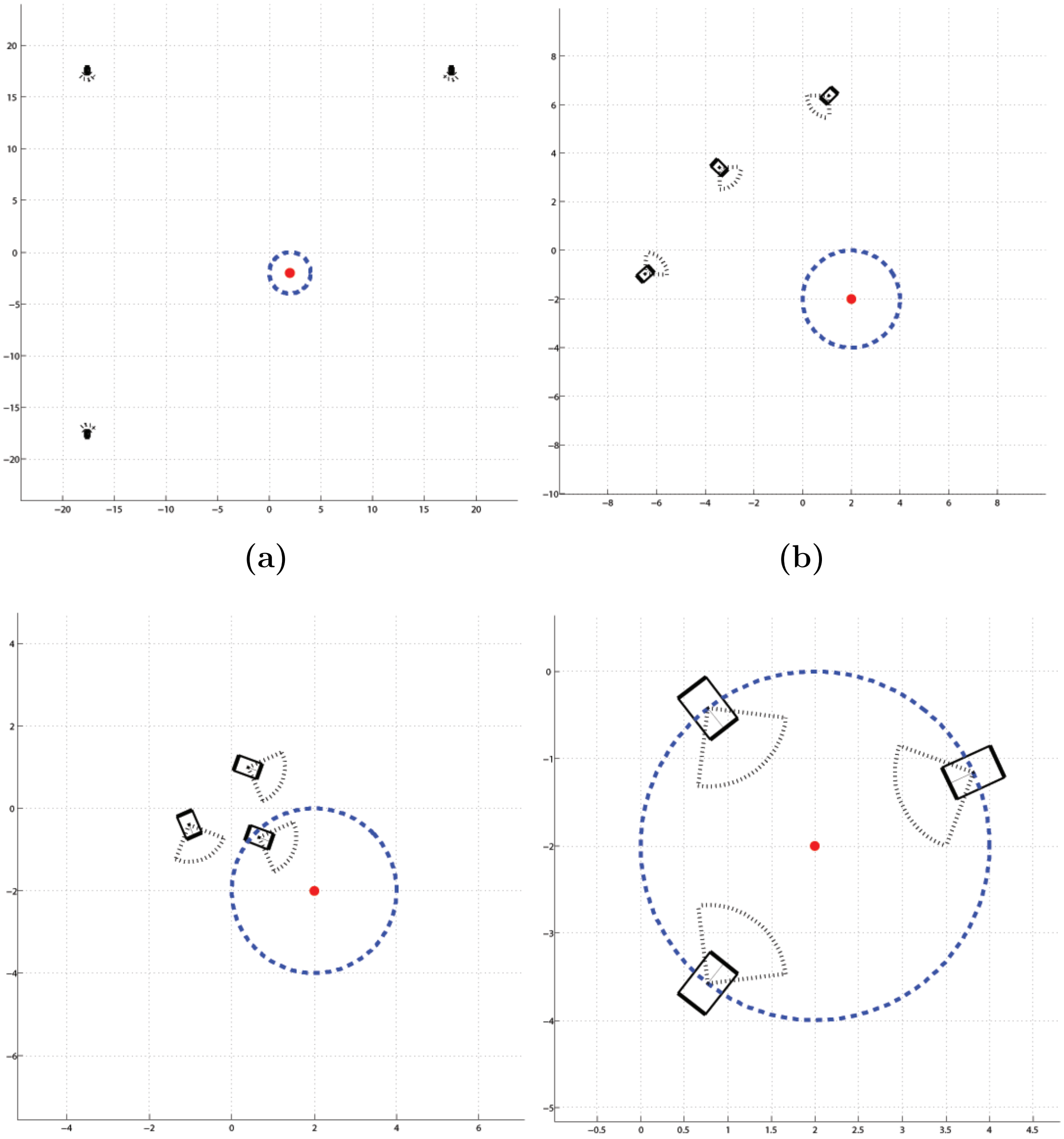


Figure 5.12: Kinematic Model with Three Robots: Trajectories of UAV and three UGVs on X-Y plane.

As it can be seen from Figure 5.13(d) the group achieves the desired formation in the final configuration. The flight information and given reference values of the quadrotor can be seen in Figure 5.14 and Figure 5.15, orientation and position, respectively.



(a) **(b)**

(c) **(d)**

Figure 5.13: Kinematic Model with Three Robots: (a) Initial configuration, (b) Zoomed version of coordinated motion, (c) Zoomed version of starting to surround T, (d) Zoomed version of desired formation.

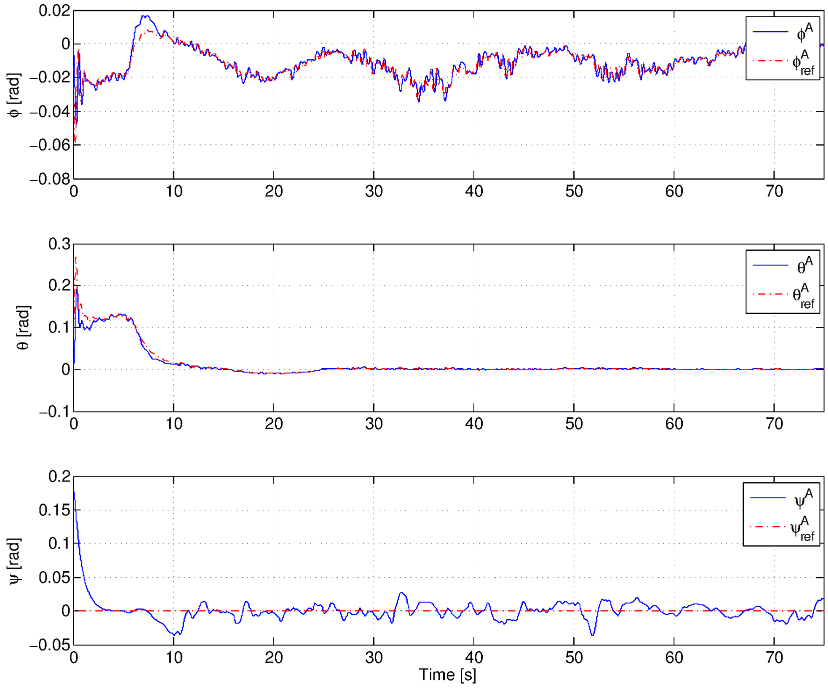


Figure 5.14: ϕ, θ, ψ of UAV with calculated reference values.

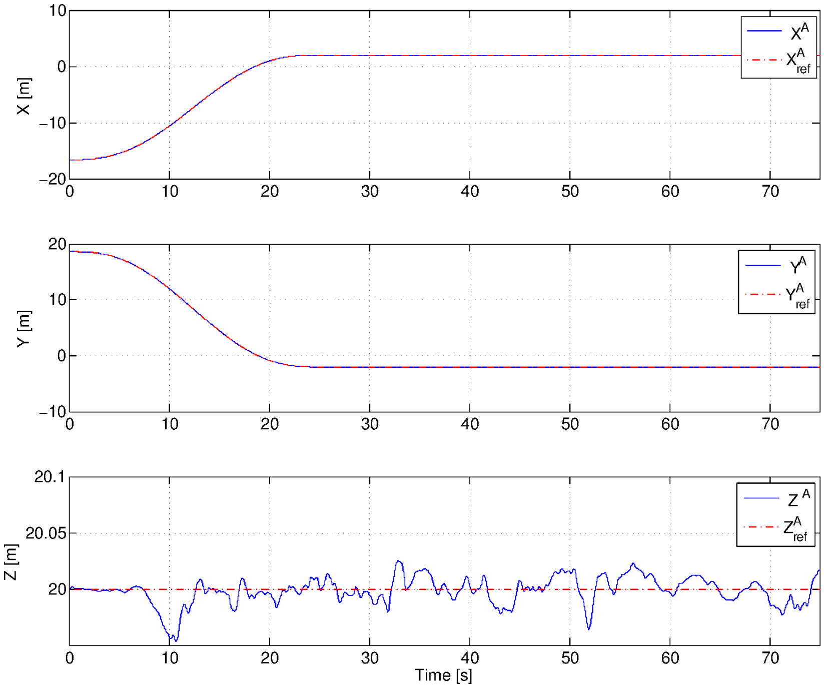


Figure 5.15: X, Y, Z of UAV with reference values.

UAV and 5 UGVs

The 3-D trajectories for UGVs and UAV can be seen in the Figure 5.16. The trajectories of UGVs and UAV are depicted on the horizontal plane in Figure 5.17. The initial setting is as in Figure 5.18(a). Since there are five UGVs, the risk of collision increases for the same value of d_T; some collisions were predicted around the formation circle, but they were successfully avoided. After UAV reaches on top of the T, the UGVs received a signal that allowed them to park. Since UAV is much faster than the UGVs, UAV hovers on. The final formation is successfully achieved with a uniform distribution of UGVs as shown in Figure 5.18(d).

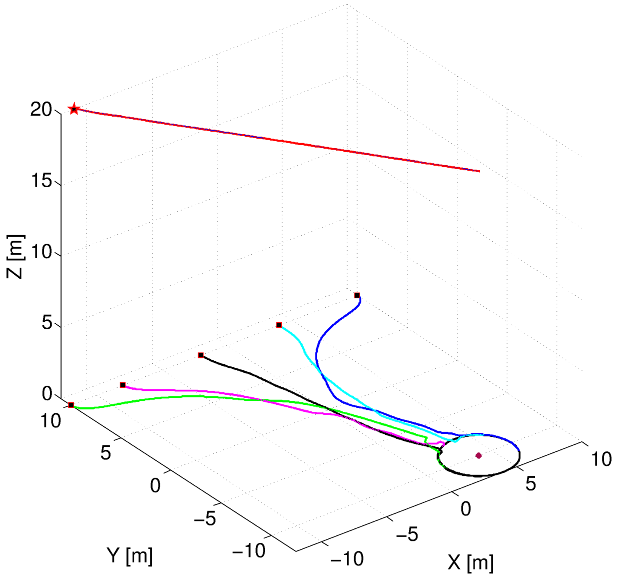


Figure 5.16: Kinematic Model with Five Robots: Trajectories of UAV and five UGVs in 3D view.

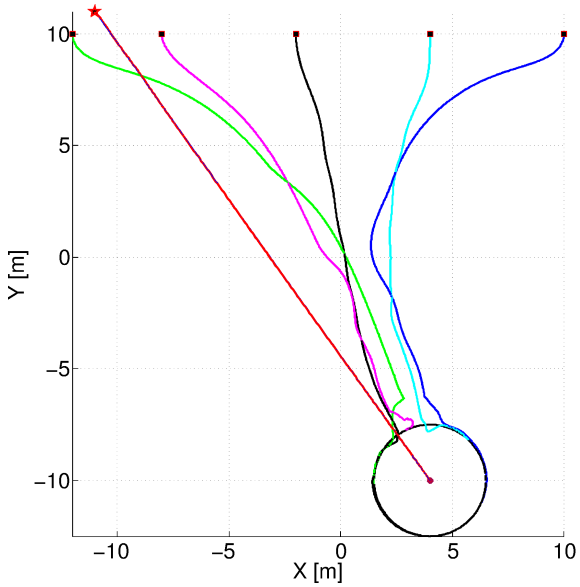
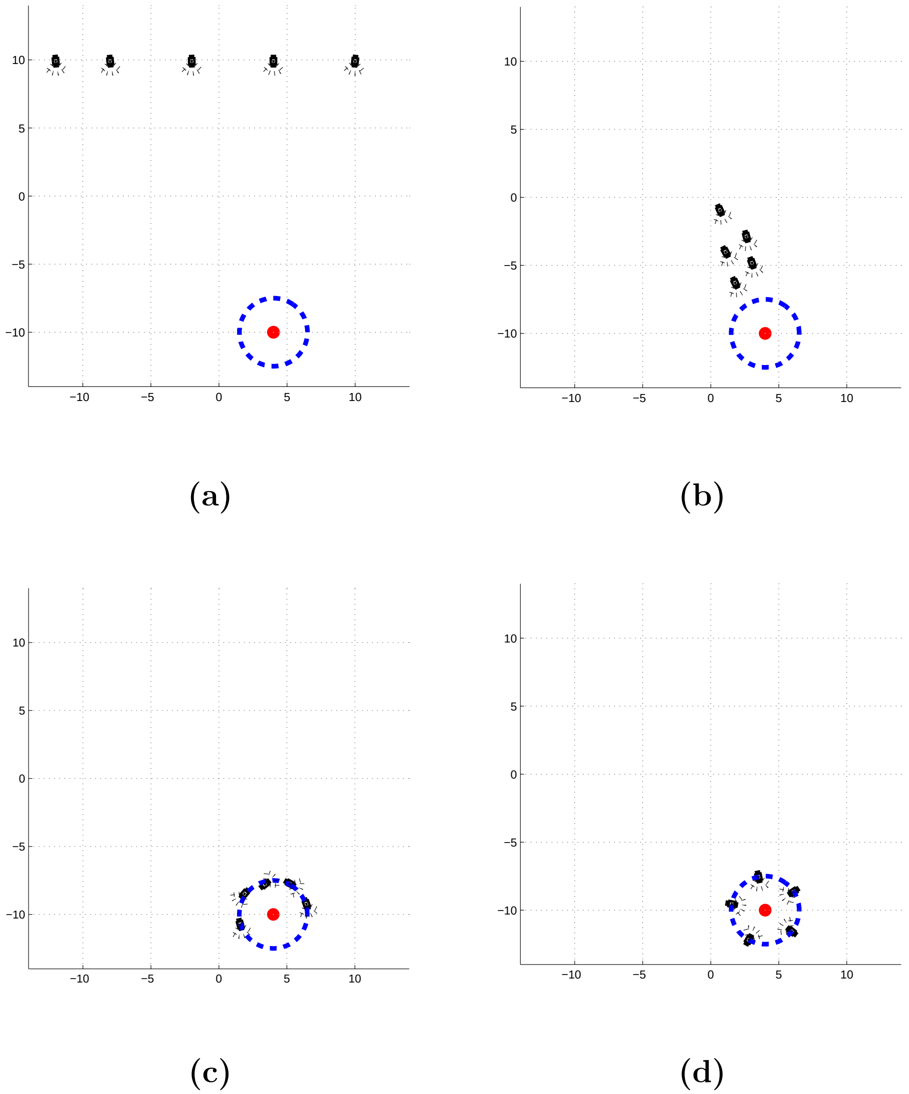


Figure 5.17: Kinematic Model with Five Robots: Trajectories of UAV and five UGVs in 2-D view, on X-Y plane.



(a)

(b)

(c)

(d)

Figure 5.18: Kinematic Model with Five Robots: (a) Initial configuration, (b) Coordinated motion, (c) Circular motion around target, (d) Desired formation.

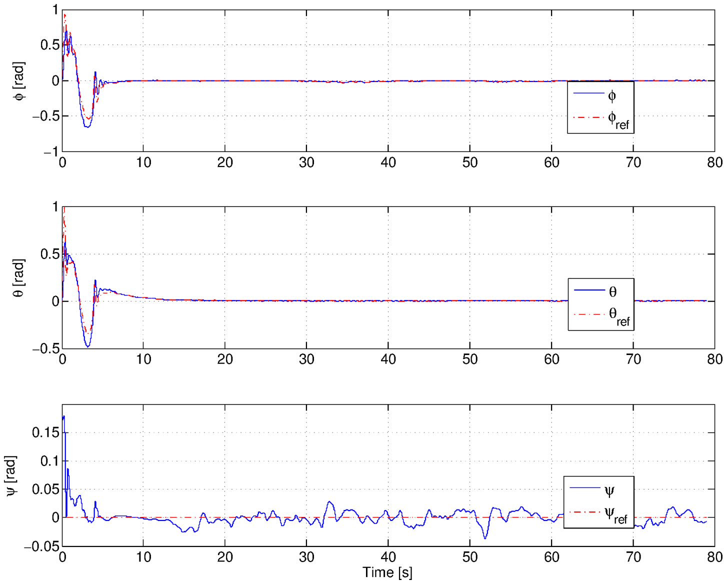


Figure 5.19: ϕ, θ, ψ of UAV with calculated reference values.

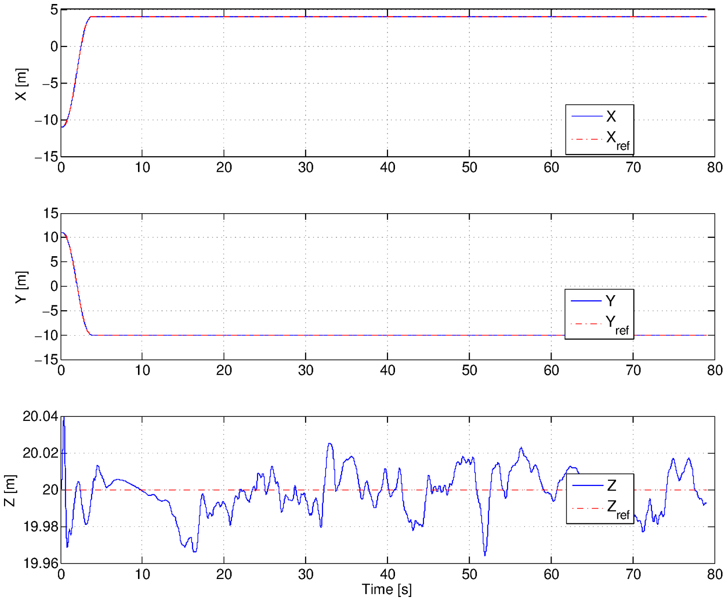


Figure 5.20: X, Y, Z of UAV with reference values.

Calculated reference values and actual values of the quadrotor's orientation can be viewed in Figure 5.19. The position references and actual positions are depicted in Figure 5.20.

5.2.2 Simulations with via-point trajectory

To test the limits of our second proposed algorithm, a via-point trajectory is used in MATLAB/Simulink with simulations and animations. In this simulations maximum angular speed for each UGV was set to $\frac{\pi}{3}$ rad/s while maximum linear speed was 1 m/s. UGVs are assumed to be motionless and UAV is assumed to be in hover at the beginning of simulations. UAV broadcasts its position information to the UGVs at 1 Hz and a 500 ms communication delay is assumed. We also added 100 ms communication delay between the UGVs. The speed of UAV is limited to 1.2 times the speed of UGVs . We have simulations for groups of three and five UGVs to collaborate with UAV in order to find and surround the T.

UAV and 3 UGVs with multiple via point trajectory

In the initial setting, the UGVs are placed as a column at the bottom left corner of a large rectangular area while T is at the upper right corner. It was observed that initially UGVs group and move towards T in a coordinated manner, under the guidance of UAV. The UAV detects the obstacles at the beginning and generates via points to create reference trajectory. To successfully guide the UGVs in the environment with obstacles UAV moves slowly, but still faster than the UGVs, till it reaches T. The UAV waits over T while broadcasting the position of the V_L. After that, when the UGVs are close enough to T they start to spread, in order to achieve neighboring distances of d_{near}. The 3-D trajectories for UGVs and UAV can be seen in the Figure 5.21. The trajectories of UGVs and UAV are depicted on the horizontal plane in Figure 5.22. As it can be seen from Figure 5.23 and Figure 5.24 the group achieves the desired formation at the final configuration. The flight information and given reference values of the quadrotor can seen in Figure 5.25 and Figure 5.26, orientation and position respectively.

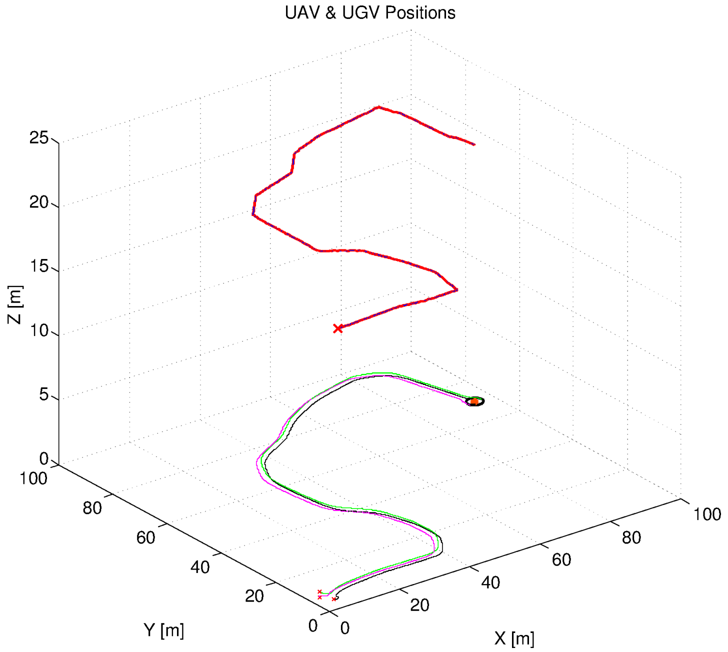


Figure 5.21: Kinematic Model with Three Robots: Trajectories of UAV and three UGVs in 3D view.

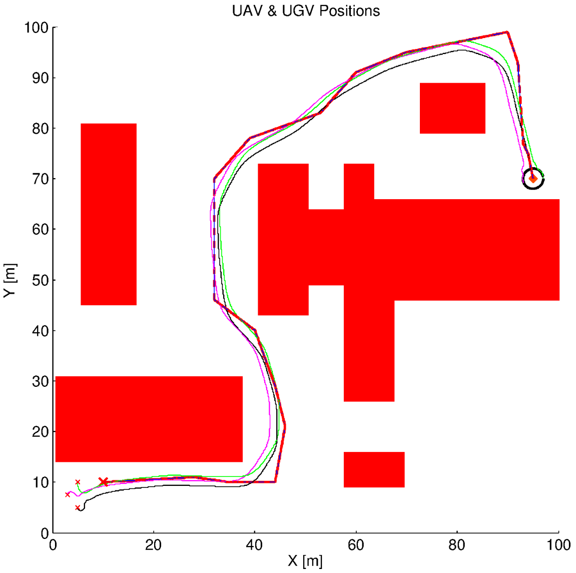


Figure 5.22: Kinematic Model with Three Robots: Trajectories of UAV and three UGVs on X-Y plane.

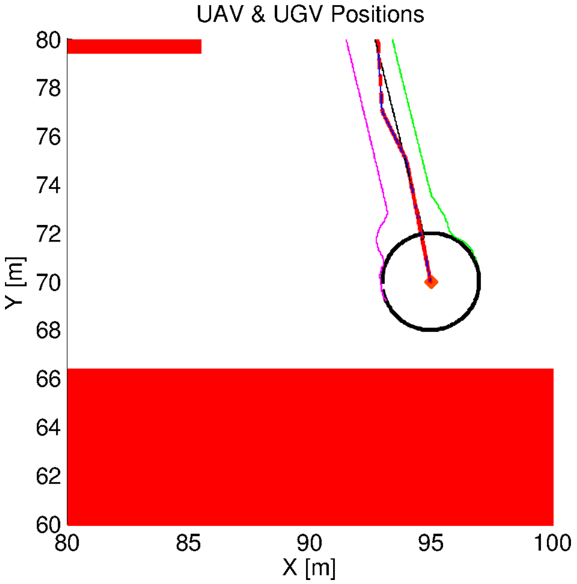


Figure 5.23: Kinematic Model with Three Robots: Zoomed final position of UAV and three UGVs on X-Y plane.

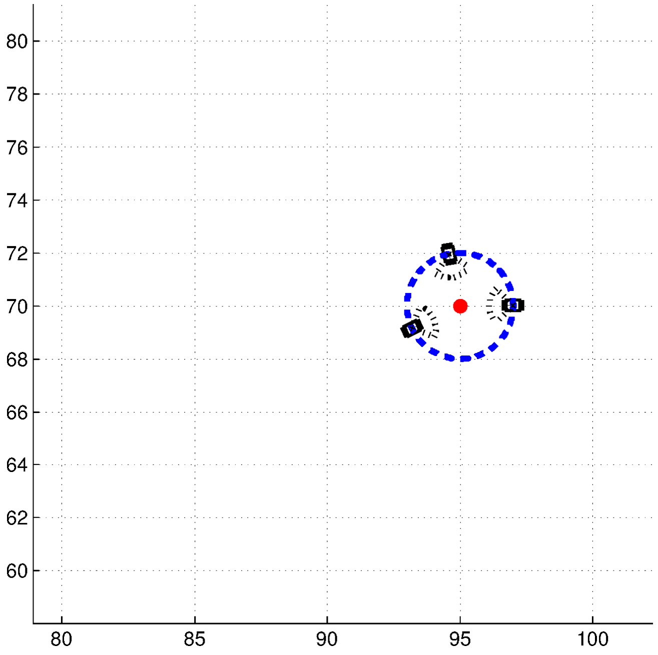


Figure 5.24: Kinematic Model with Three Robots: Zoomed final pose of UAV and three UGVs on X-Y plane.

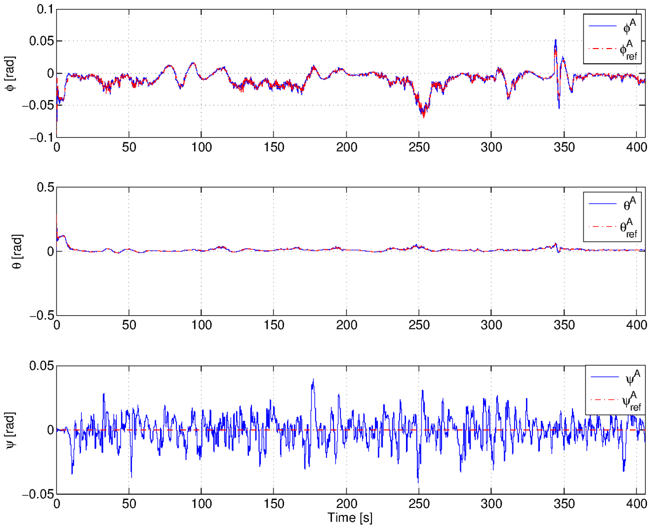


Figure 5.25: ϕ, θ, ψ of UAV with calculated reference values.

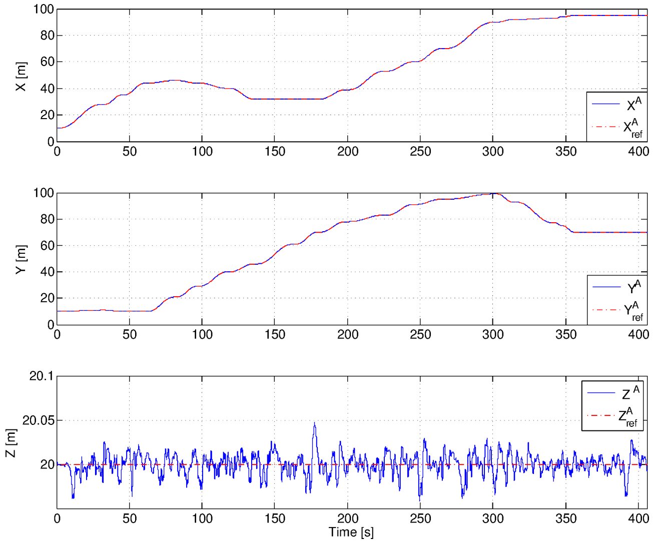


Figure 5.26: X, Y, Z of UAV with reference values.

UAV and 5 UGVs with multiple via point trajectory

Similar to the initial setting for three UGV simulation, the five UGVs are placed at the bottom left corner of a rectangular area while T is at the upper right corner. The UGVs are successfully towards T guided under the supervision of UAV. The UAV detects the obstacles at the beginning and generates via points to create reference trajectory. To successfully guide the UGVs in the environment with obstacles UAV speed is limited. The UAV moves faster than the UGVs, and when it reaches T it starts hover on top of T while broadcasting the position of the V_L. UGVs start to spread when they are close enough to T, in order to achieve neighboring distances of d_{near}. Since there are five UGVs, the risk of collision increases for the same value of d_T; some collisions were predicted around the formation circle, but they were successfully avoided. The 3-D trajectories for UGVs and UAV is depicted in the Figure 5.27. The trajectories of UGVs and UAV can be seen on the horizontal plane in Figure 5.28. Finally, in Figure 5.29 and in Figure 5.30 it can be seen that the group achieves the desired formation as their final configuration.

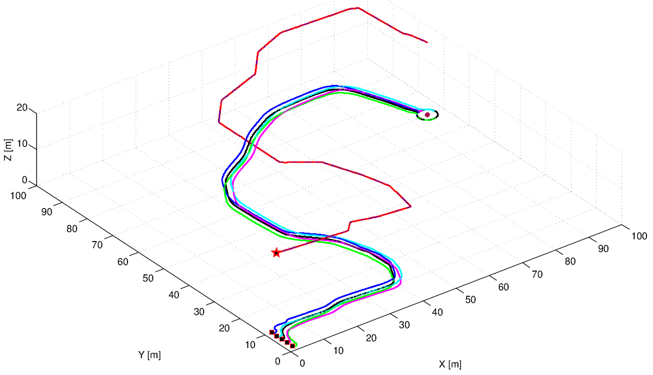


Figure 5.27: Kinematic Model with Five Robots: Trajectories of UAV and five UGVs in 3D view.

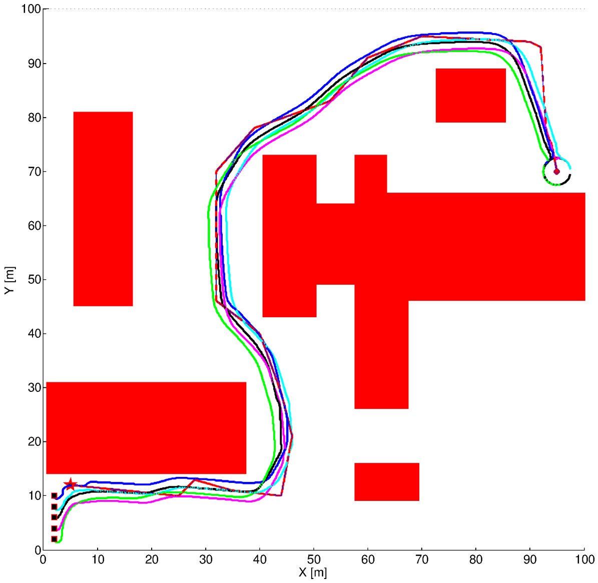


Figure 5.28: Kinematic Model with Five Robots: Trajectories of UAV and five UGVs on X-Y plane.

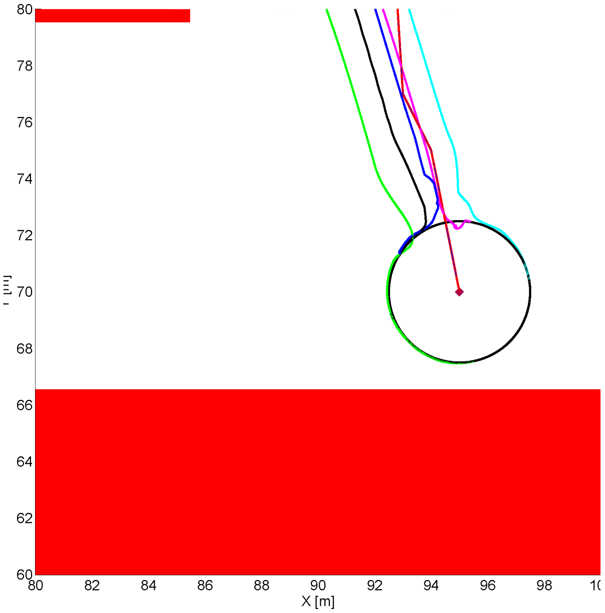


Figure 5.29: Kinematic Model with Five Robots: Zoomed final position of UAV and five UGVs on X-Y plane.

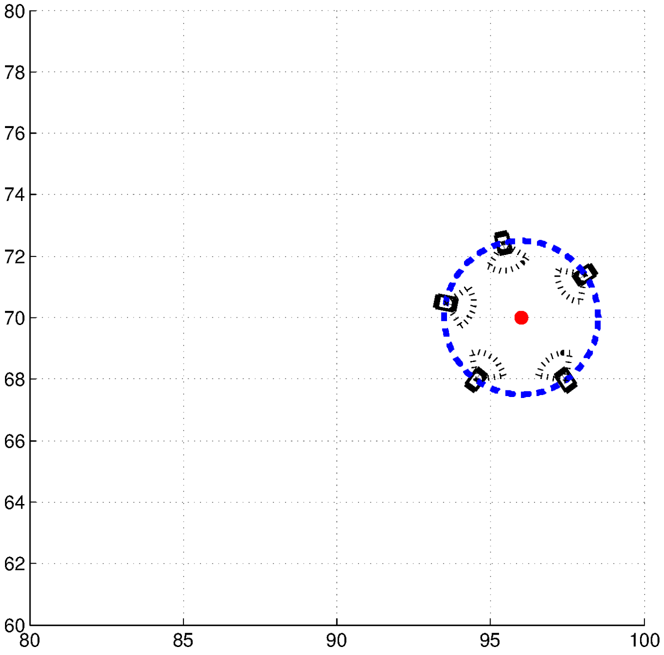


Figure 5.30: Kinematic Model with Five Robots: Zoomed final pose of UAV and five UGVs on X-Y plane.

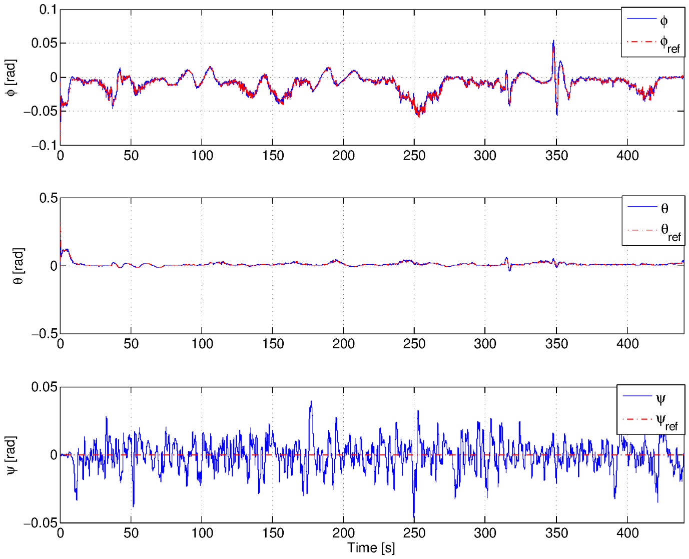


Figure 5.31: ϕ, θ, ψ of UAV with calculated reference values.

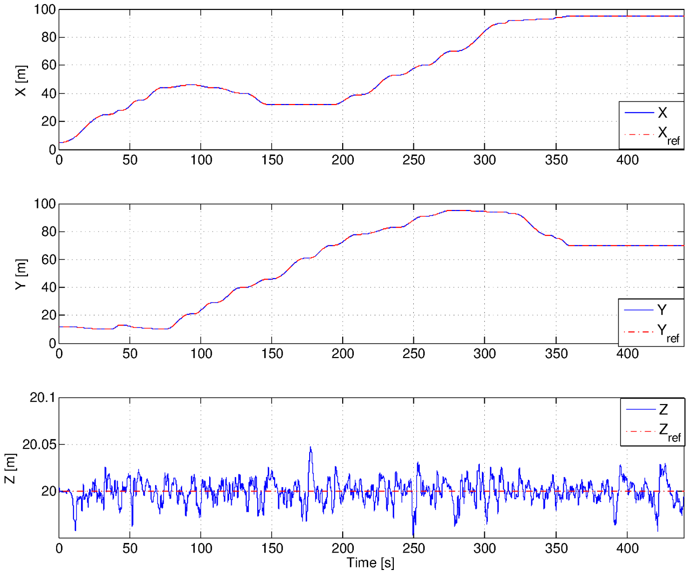


Figure 5.32: X, Y, Z of UAV with reference values.

Chapter 6

Conclusions and Future Work

In this thesis, two novel coordination models for a group of unmanned ground vehicles guided by an unmanned aerial vehicle have been proposed. In both methods, a decentralized coordination algorithm and a virtual leader is employed in order to coordinate a group of UGV by a UAV. The main difference in these two models is the coordination logic between UGVs. The UGV considered in this thesis is a two-wheeled nonholonomic mobile robot, while the UAV is a quadrotor type aerial vehicle. The UGVs are assumed to be able to communicate their positions and velocities to each other by some communication protocol, while the orientation information is calculated by considering the nonholonomic constraints. Possible collisions between UGVs are predicted and avoided by an algorithm that was developed previously. The UAV is capable of detecting the target and producing the map of the environment which is going to be used for creating an appropriate path, by a path planning algorithm before operation starts.

After detecting the target, UAV flies from an initial point, which is close to UGVs, to the top of the target. In order to do so a hierarchical control scheme is preferred for the flight controllers of UAV. Flight controller is divided into two controllers, attitude and position controllers respectively. The reference for attitude controller is generated dynamically from position reference.

In the first method a virtual mass spring damper system is introduced between the two closest neighbour of a UGV and the V_L as it can be seen in Fig. 3.2. This dynamical system is used for generating the references for each UGV. The main drawback of this system is the relaxation of the nonholonomic constraints. Because of this reason the system might become stiff. The first method is not tested in the environment with obstacles but verified for simple trajectories. The results shown in the previous chapter are still promising.

To improve the coordination between UAV and UGVs a second method is proposed. In this method to satisfy the nonholonomic constraints of UGVs we have defined virtual reference robots that enabled us to create trackable reference pose for each robot from the linear and angular velocities. In order to increase the degree of freedom of the system several adaptive parameters are included into the algorithm. The effect of the V_L is also included in this system and UGVs are guided to a desired location and then surround the target.

As it can be seen from the simulation results, we are able to navigate a group of UGVs by a UAV, to the vicinity of T in a coordinated manner by using both methods. The goals defined in 3.1 are achieved successfully. The second method is also verified in the environment with obstacles. In our simulations we also introduced 500 ms communication delay between UGVs and UAV, and 100 ms delay between UGVs.

As future work, we are planning to work on the physical implementation of the proposed coordination scheme with nonholonomic mobile robots and a quadrotor in order to surround and manipulate an object. Also the proposed algorithm can be extended to perform search and rescue type of tasks or surveillance and reconnaissance type of missions. Another path for extending this work is employing different communication schemes in favor of reducing the bandwidth load of the system. Similarly the performance of the system can be inspected for the cases where one or more robots malfunction.

Bibliography

[1] Dynastat memory aids unimate robot. *Electrical Engineering*, 80(4):319–320, 1961.

[2] G.E. Munson. Robots quietly take their places alongside humans on the production line to raise productivity - and do the "dirty work". *Spectrum, IEEE*, 15(10):66–70, 1978.

[3] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 500–505, 1985.

[4] B. Kwolek, T. Kapuscinski, and M. Wysocki. Vision-based implementation of feedback control of unicycle robots. In *Proceedings of the First Workshop on Robot Motion and Control. RoMoCo '99.*, pages 101–106, 1999.

[5] 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 Conference on Decision and Control*, volume 3, pages 2926–2931 Vol.3, 2003.

[6] Joseph Egbert and Randal W. Beard. Low-altitude road following using strap-down cameras on miniature air vehicles. *Mechatronics*, 21(5):831 – 843, 2011. Special Issue on Development of Autonomous Unmanned Aerial Vehicles.

[7] International Federation of Robotics. History of industrial robots;from the first installation until today. `http://www.ifr.org/uploads/media/History_of_Industrial_Robots_online_brochure_by_IFR_2012.pdf`, 2012. Accessed: 2013-06-25.

[8] Stanford Research Institute. Shakey. `http://www.ai.sri.com/shakey/`, 2013. Accessed: 2013-06-25.

[9] H. Yamaguchi. A cooperative hunting behavior by mobile-robot troops. *The International Journal of Robotics Research*, 18(9):931–940, 1999.

[10] Tamas Vicsek, Andras Czirok, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1229, Aug 1995.

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

[12] L. Merino, F. Caballero, J.R.M.-d. Dios, and A. Ollero. Cooperative fire detection using unmanned aerial vehicles. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1884–1889, April.

[13] J. Tisdale, A. Ryan, Zu Kim, D. Tornqvist, and J.K. Hedrick. A multiple uav system for vision-based search and localization. In *American Control Conference, 2008*, pages 1985–1990, June.

[14] M. Saska, V. Vonasek, T. Krajnik, and L. Preucil. Coordination and navigation of heterogeneous uavs-ugvs teams localized by a hawk-eye approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'12*, pages 2166–2171, 2012.

[15] Intel. Moore's law 40th anniversary. `http://www.intel.com/pressroom/kits/events/moores_law_40th/`, 2013. Accessed: 2013-06-25.

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

[17] A. Chikwanha, S.Motepe, and R. Stopforth. Survey and requirements for search and rescue ground and air vehicles for mining applications. In *19th International Conference Mechatronics and Machine Vision in Practice (M2VIP)*, pages 105–109, 2012.

[18] iRobot. irobot roomba® vacuum cleaning robot. `http://www.irobot.com/en/us/learn/home/roomba.aspx`, 2013. Accessed: 2013-07-02.

[19] M.A. Klein, A. S. Boxerbaum, R. D. Quinn, R. Harkins, and R. Vaidyanathan. Seadog: A rugged mobile robot for surf-zone applications. In *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, pages 1335–1340, 2012.

[20] United States Department of Defence. Unmanned systems integrated roadmap. `http://www.defenseinnovationmarketplace.mil/resources/UnmannedSystemsIntegratedRoadmapFY2011.pdf`, 2011. Accessed: 2013-06-25.

[21] Y. Kim, J. Kang, D. Kim, E. Kim, P. K. Chong, and S. Seo. Design of a fence surveillance system based on wireless sensor networks. In *Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems*, pages 4:1–4:7, Brussels, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[22] United States Department of Defence. Unmanned systems roadmap 2007-2032. `https://www.fas.org/irp/program/collect/usroadmap2007.pdf`, 2007. Accessed: 2013-06-25.

[23] M.B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multi-robot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[24] X. Yuan and S. X. Yang. Multi-robot coordination with balanced task allocation and optimized path planning. In *IEEE International Conference on Robotics and Biomimetics, 2007, ROBIO 2007.*, pages 1007–1011, 2007.

[25] Hamido Hourani, Philipp Wolters, Eckart Hauck, and Sabina Jeschke. A marsupial relationship in robotics: A survey. In Sabina Jeschke, Ingrid Isenhardt, Frank Hees, and Klaus Henning, editors, *Automation,*

Communication and Cybernetics in Science and Engineering 2011/2012, pages 667–679. Springer Berlin Heidelberg, 2013.

[26] M. Lindemuth, R. Murphy, E. Steimle, W. Armitage, K. Dreger, T. Elliot, M. Hall, D. Kalyadin, J. Kramer, M. Palankar, K. Pratt, and C. Griffin. Sea robot-assisted inspection. *Robotics Automation Magazine, IEEE*, 18(2):96–107, 2011.

[27] Simon Lacroix and Guy Besnerais. Issues in cooperative air/ground robotic systems. In Makoto Kaneko and Yoshihiko Nakamura, editors, *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 421–432. Springer Berlin Heidelberg, 2011.

[28] Service Robots Group Universita di Catania. Unmanned aerial vehicle for flying over volcanoes. `http://www.robotic.diees.unict.it/robots/uav/uav.htm`, 2013. Accessed: 2013-07-10.

[29] Department of the Navy Office of Naval Research. Ad hoc networks today. `http://img.docstoccdn.com/thumb/orig/104670159.png`, 2013. Accessed: 2013-07-10.

[30] R. Hudjakov and M. Tamre. Aerial imagery terrain classification for long-range autonomous navigation. In *International Symposium on Optomechatronic Technologies, ISOT2009.*, pages 88–91, 2009.

[31] C. Luo, A.P. Espinosa, A. D. Gloria, and R. Sgherri. Air-ground multi-agent robot team coordination. In *IEEE International Conference on Robotics and Automation, ICRA'11*, pages 6588–6591, 2011.

[32] S. Ioannou, K. Dalamagkidis, K.P. Valavanis, and E.K. Stefanakos. Improving endurance and range of a ugv with gimballed landing platform

for launching small unmanned helicopters. *Journal of Intelligent and Robotic Systems*, 53(4):399–416, 2008.

[33] N. Giakoumidis, J. U. Bak, J. V. Gomez, A. Llenga, and N. Mavridis. Pilot-scale development of a uav-ugv hybrid with air-based ugv path planning. In *10th International Conference on Frontiers of Information Technology, FIT 2012*, pages 204–208, 2012.

[34] M. H. Mahoor, R. Godzdanker, K. Dalamagkidis, and K.P. Valavanis. Vision-based landing of light weight unmanned helicopters on a smart landing platform. *Journal of Intelligent & Robotic Systems*, 61(1-4):251–265, 2011.

[35] M. Saska, T. Krajnik, and L. Pfeucil. Cooperative μ UAV-UGV autonomous indoor surveillance. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 1–6, 2012.

[36] H. G. Tanner. Switched uav-ugv cooperation scheme for target detection. In *IEEE International Conference on Robotics and Automation*, pages 3457–3462, 2007.

[37] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas. Cooperative air and ground surveillance. *Robotics Automation Magazine, IEEE*, 13(3):16–25, 2006.

[38] C. Phan and H. H. T. Liu. A cooperative uav/ugv platform for wildfire detection and fighting. In *Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing, ICSC 2008.*, pages 494–498, 2008.

[39] C. Hager, D. Zarzhitsky, K. Hyukseong, and D. Pack. Cooperative target localization using heterogeneous unmanned ground and aerial vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2010*, pages 2952–2957, 2010.

[40] R. Siegwart and I. R. Nourbakhsh. *Introduction To Autonomous Mobile Robots*. The MIT Press, 2004.

[41] E.T. Baumgartner and S.B. Skaar. An autonomous vision-based mobile robot. *IEEE Transactions on Automatic Control*, 39(3):493–502, 1994.

[42] N. Gulec and M. Unel. A novel coordination scheme applied to nonholonomic mobile robots. In *44th IEEE Conference on Decision and Control and European Control Conference. CDC-ECC'05*, pages 5089–5094, 2005.

[43] D. Vissiere, D. E. Chang, and N. Petit. Experiments of trajectory generation and obstacle avoidance for a ugv. In *American Control Conference. ACC'07*, pages 2828–2835, 2007.

[44] C. Samson and K. Ait-Abderrahim. Feedback stabilization of a nonholonomic wheeled mobile robot. In *Proceedings IEEE/RSJ Intelligent Robots and Systems '91, International Workshop on 'Intelligence for Mechanical Systems, IROS '91*, pages 1242–1247 vol.3, 1991.

[45] Mingcong Deng, A. Inoue, and K. Sekiguchi. Parking control of a two wheeled mobile robot. In *International Conference on Mechatronics and Automation, ICMA 2007*, pages 539–544, 2007.

[46] A. Khoukhi. An intelligent multi-agent system for mobile robots navigation and parking. In *IEEE International Workshop on Robotic and Sensors Environments ,ROSE*, pages 1–6, 2010.

[47] C. Samson and K. Ait-Abderrahim. Feedback control of a nonholonomic wheeled cart in cartesian space. In *IEEE International Conference on Robotics and Automation*, pages 1136–1141 vol.2, 1991.

[48] P. Morin and C. Samson. Trajectory tracking for non-holonomic vehicles: overview and case study. In *Proceedings of the Fourth International Workshop onRobot Motion and Control. RoMoCo'04.*, pages 139–153, 2004.

[49] E. Cetinsoy, S. Dikyar, C. Hancer, K.T. Oner, E. Sırımoglu, M. Unel, and M. F. Aksit. Design and construction of a novel quad tilt-wing uav. *Mechatronics*, 22(6):723–745, Sept. 2012.

[50] L. Chaimowicz, A. Cowley, D. Gomez-Ibanez, B. Grocholsky, M.A. Hsieh, H. Hsu, J.F. Keller, V. Kumar, R. Swaminathan, and C.J. Taylor. Deploying air-ground multi-robot teams in urban environments. In *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, pages 223–234. Springer Netherlands, 2005.

[51] Tommaso Bresciani. Modelling , identification and control of a quadrotor helicopter. Master's thesis, Department of Automatic Control, Lund University, 2008.

[52] F. Nonami, K.and Kendoul, S. Suzuki, W. Wang, and D. Nakazawa. *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles.* Springer, Japan, 2010.

[53] Nusrettin Güleç. Modeling and control of the coordinated motion of a group of autonomous mobile robots. Master's thesis, Sabancı University, Istanbul, Turkey, 2005.

[54] Nusrettin Güleç and Mustafa Ünel. A novel algorithm for the coordination of multiple mobile robots. In *International Symposium on Computer and Information Sciences. ISCIS 2005*, volume 3733 of *Lecture Notes in Computer Science*, pages 422–431. Springer Berlin Heidelberg, 2005.

[55] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics. Springer, 2011.