1-1-2019

Distributed Formation Control for Ground Vehicles with Visual Sensing Constraint

Jie Tang
University of Windsor

Distributed Formation Control for Ground Vehicles with Visual Sensing Constraint

by

Jie Tang

A Thesis

Submitted to the Faculty of Graduate Studies

through the Department of Electrical and Computer Engineering in Partial

Fulfillment of the Requirements for

the Degree of Master of Applied Science

at the University of Windsor

Windsor, Ontario, Canada

2019

Distributed Formation for Ground Vehicles with Visual Sensing Constraint

by

Jie Tang

APPROVED BY:

December 4, 2019

Author's Declaration of Originality

I hereby certify that I am the sole author of this major paper and that no part of this major paper has been published or submitted for publication.

I certify that, to the best of my knowledge, my major paper does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my major paper, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s)in my major paper and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my major paper, including any final revisions, as approved by my major paper committee and the Graduate Studies office, and that this major paper has not been submitted for a higher degree to any other University or Institution.

Abstract

Formation control combined with different tasks enables a group of robots to reach a geographical location, avoid a collision, and simultaneously maintain the designed formation pattern. The connection and perception are critical for a multi-agent formation system, mainly when the robots only use vision as a communication method. However, most visual sensors have limited Field-of-view (FOV), which leaves some blind zones. In this case, a gradient-based distributed control law can be designed to keep every robot in the visible zones of other robots during the formation. This control strategy is designed to be processed independently on each vehicle with no network connection. This thesis assesses the feasibility of applying the gradient descent method to the problem of visual constraint vehicle formation.

Keywords: Formation Control, Distributed Control, Visual Constraint.

Dedication

Dedicated to my beloved parents and my lovely girlfriend, Weilin Fang.

Acknowledgments

I would like to take this opportunity to thank my supervisor, Dr. Xiang Chen, for his elaborate guidance through every stage of this research paper. His support and feedback helped in transforming this paper into a more meaningful one. His immense passion for teaching and research have further inspired me to pursue this path in the future.

I would also like to thank my committee members, Dr. Balakumar Balasingam and Dr. Jalal Ahamed, for their constructive comments, valuable feedback, positive criticism and their time in reviewing my work.

I would like extend my gratitude to my lab colleagues, Tong Zhang and Youying Hua, in the Graduate Control and Robotics Lab, for their friendship, support, their constant involvement, and their valuable feedback.

Finally, I would like to thank my family and friends who continuously aid me in my research. My parents, Jun Tang and Wenhui Zhang, support me both mentally and financially, and my friend Zichun Zhao helped me do the experiments on weekends and holidays.

Contents

List of Tables

List of Figures

List of Abbreviations

FOV	Field of View
HD	High Definition
IMU	Inertial Measurement Unit
MAE	Mean Absolute Error
MATLAB	Matrix Laboratory
MCU	Micro Controller Unit
NVIDIA	NVIDIA Corporation
PID	Proportional, Integral, and Derivative
PnP	Perspective-n-Point
ROS	Robot Operation System
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol

Chapter 1

Introduction

A group of robots needs cooperation when fulfilling a certain task. It has numerous reasons to let robots work cooperatively. Not only efficiency, redundancy, and flexibility can be improved dramatically, but also, some tasks single robot impossibly performed are feasible with multi-robots. Although the prospect is attractive, there are still some scientific and technological challenges. One can imagine the control of a group of robots, the method of communications among multi-robots, dealing with unstable and constraint communicating, etc. These kinds of challenges will be discussed in following sections, section.1.1 and section.1.2. Also, some popular control strategies of multi-robots cooperation, particularly formation control, will be introduced in section.1.3.

1.1 Vision-based Formation of Ground Vehicles

Formation control generally refers to the control approach to accomplish a specific pattern with a group of robots. It is the foundation of autonomous cooperative control with multi-robots, which can be widely used in surveillance, distributed manipulation, mapping of unknown environments, and transportation of large objects[3]. Formation control includes research from a wide variety of fields such as computer vision, sensor fusion, embedded systems, and distributed systems.

Meanwhile, depending on whether there is any central controller doing motion planning and making decisions for all the vehicles, a formation control can be classified to the central or distributed controller. Distributed control strategy means that the robots give a response mainly relay on the local information and decisions. The central operator may exist but only fulfill the supervisory control. Excepting with the feature of full autonomy, the distributed system also has the following benefits: (1)The computing can be done parallel. (2)The system will be more robust without the central controller. (3)The increase of total agent numbers will not dramatically rise the computing power.

One of the critical aspects of a robotic system is perception. It determines how robots senses the environment and extract useful information, which usually means the estimation of robots' pose for the formation problem. Vision has long been considered as one of the most intuitional and powerful means of perception. Visual sensing can provide sufficient information to perform control tasks. However, the challenge will come with the fact that real-time control will require a lot of computing power, especially with distributed onboard computation. Also, the limitation of field of view is another challenge, which will be discussed later in the next section.

In the 2D case, the ground vehicles can be chosen as an appropriate research object. Most of the ground vehicles have nonholonomic kinematics, which constrains the movement along the side direction. The nonholonomic constraints should also be take into consideration when designing the control algorithm.

1.2 Field of View Constraint

It is well-known that every sensor has a limitation with the working range and field. As for vision sensors, the field-of-view (FOV) defines the area of inspection captured by the camera. It restricts the horizontal and vertical angle as a boundary for visual sensors. Under the condition of limited FOV constraint, the visual connection between every two robots may be unstable during the formation. In some sensing and communication constraint cases, one of the tasks could be maintaining visibility among all the vehicles.

In some particular scenarios, the robots need to see all each other simultaneously. For example, a group of UAVs or robots with bidirectional full vision connections can prepare defense when any vehicle in formation is getting attacked by the enemy or invader. Keeping visibility is meaningful for defense-purposed surveillance.

1.3 Literature Review

In recent years, our society has developed an ever-growing interest in autonomous cooperative agents. Specifically, formation problem has been attracting the most attention in the multi-agent research field. Based on various criteria, some control architectures have been constructed in the past few years, such as leader-follower [1,4,5], virtual structure [2,6,7], behavioural-based [8] etc.

In the leader-follower approach [1], the leader and follower relationships between two vehicles are defined. The leaders track the predefined or on-line planned trajectories, and the followers track leaders based on the relative pose of their nearby leaders. The leaders do not need to maintain the formation and have more flexibility compared with the followers. Concurrently, the first follower will attempt to maintain the distance and the orientation with the leaders. Meanwhile, the second follower will try to track the first follower in the front . The leader-follower approach is easy to implement and understand. Besides, when leaders are disturbed by external environments, the formation can still maintain the objective pattern. However, in the leader-follower approach, the leader vehicles do not have the feedback information from follower vehicles. A general illustration of the leader-follower approach is shown in Fig.1.1.

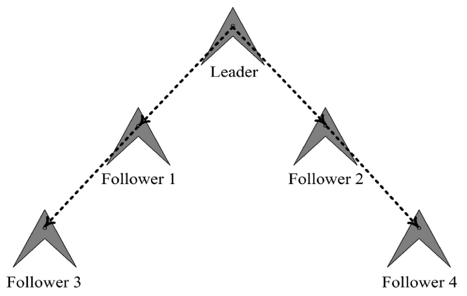


Figure 1.1: A Leader-follower Control Strategy[1]

In[2] a concept of virtual structure is introduced. The author uses virtual structure to force an ensemble of robots to behave as if they form a rigid body. The control strategy for virtual structure can be illustrated as a *bi-direction control*. The mobile robots can be controlled by applying a virtual force field to the whole structure; also the position and shape of the virtual structure is determined by the action of each mobile robots. Many considerable works based on virtual structure have been done in the past few years. In[6], Norman and Hugh decrease the formation error by using motion synchronization to modify the trajectory. In[7], formation feedback is used to improve the robustness of virtual structure formation. The advantage of this approach is that it is easy to maneuver robots with group behaviour. However, it is challenging to do extra applications when the formation needs to maintain the virtual structure all the time. Also, the synchronization cost may be high and frequent to keep precise formation patterns. A brief illustration of the virtual structure method is shown in Fig.1.2

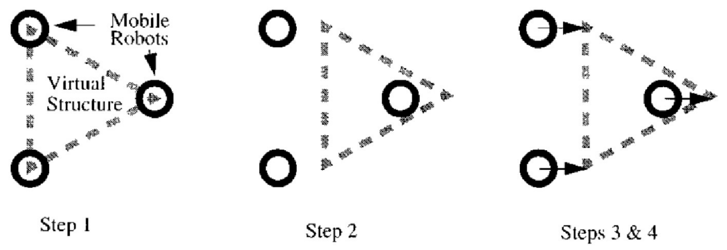


Figure 1.2: The Virtual Structure Control Strategy[2]

A behavior-based approach is introduced by considering of doing various tasks, such as avoiding an obstacle, tracking neighbor vehicles, and formation keeping, simultaneously during the formation. Tucker, and Ronald[8] used the gain factors to balance the importance of different objectives. The combined behavior was generated by multiplying the outputs of each fundamental objective by its weighting gain, then summing and normalizing the results. In[9], the initial formation problem combined with the navigation task was solved by extending the behavior-based approach with a navigating algorithm. It is natural to use a behavior-based approach to describe a formation problem with multiple objectives; however, this approach has disadvantage because the behavior-based approach makes it difficult to analyze the inherent mathematical properties and prove the stability and convergence.

One of the main challenges for real-time visual-based formation control is the localization of other vehicles. . Locating robots has been an active research topic in recent years[10] [11] [12]. There are currently two ways for 6D vision-based pose estimations. The first is solving PnP (Perspective-n-Point) problems[13], which is a traditional way for visual . The second is deep learning methods[14] also become popular in recent years. In the current study, the artificial feature-based marker PnP method is used for localization.

1.4 Research Objective

The main goal of this thesis is to design a distributed formation system that can be work with visual constraints. The overall design primarily consists of the vision localization and distributed local controller.

Three performances will be used as the primary evaluation criteria that will be tested in the thesis:

1. objective velocity and handing orientation synchronizing,
2. formation pattern convergence and maintenance, and
3. avoidance of blind zones.

1.5 Thesis Outline

This thesis begins with an introduction to the visual-based formation tasks and field of view (FOV) constraints in chapter 1. Section 1.3 gives some popular formation control strategies.

Chapter 2. talks about the theoretical foundations used in this thesis. Section 2.1 introduces some basic geometry knowledge that will help to characterize the formation problem. Section 2.2 gives two widely used vehicle dynamic models. Section 2.3 presents a brief introduction of graph theory, which is used to describe multi-agent problems. Section 2.4 introduces the gradient-based formation control method, which will be used to solve the related formation problem in this thesis. Section 2.5 shows the camera model and visual-based pose estimation method that facilitate localization between vehicles.

Chapter 3 introduces the design of distributed formation control system. Section 3.1 outlines the problem formulation of visual constraint formation task . Potential function design, including visual constraint C_{ij} and formation pattern V_{ij}, and local controller design are the main components of section 3.2. Section 3.3 presents a control structure of each vehicle.

Chapter 4 focuses on the details of implementing a formation control system. Section 4.1 presents the general structure of a distributed formation system , and section 4.2 shows the configuration of the vehicles used in the experiment, including the kinematics, physical performance, and localization method. Software and the library package used in the project are shown in section 4.3.

Chapter 5 shows the experiment setups and analysis of the results. Introductions of the experimental environments are given in section 5.1, while random initial pose conditions are used in section 5.2 to test the convergence ability for different initial states of the system. In section 5.3, the detailed analysis of experiments is illustrated, which includes the trajectory, speed, orientation, and relative angles.

Chapter 6 ends with conclusion and future works.

Chapter 2

Theoretical Foundations

To support the some of the mathematical definitions and technical concepts this thesis utilizes, this chapter will give a brief review of geometry definitions, dynamic models of vehicle, graph theories, and gradient control. The chapter also introduces the camera model and pose estimation method.

2.1 Geometry

2.1.1 Euclidean Distance

The Euclidean distance is the most obvious way of representing the distance between two points in a metric space. The distance between two points $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$ in n-dimensional Euclidean space is defined as:

$$\|X - Y\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.1}$$

2.1.2 Inner Product

On $\Re^2$, a Euclidean vector is a geometric object that possesses both a magnitude and a direction. Its magnitude is its length, and its direction is the direction that the arrow points to. The magnitude of a vector $\mathbf{a}$ is denoted by $\|\mathbf{a}\|$. The inner product of two Euclidean vectors $\mathbf{a}$ and $\mathbf{b}$ is defined by

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\|\|\mathbf{b}\| \cos\theta \tag{2.2}$$

Where θ is the angle between $\mathbf{a}$ and $\mathbf{b}$. It also can be calculated by

$$\theta = \arccos \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \tag{2.3}$$

2.1.3 Rotation Matrix

A rotation matrix $\mathbf{R}$ in Euclidean n-space is a $n \times n$ real orthogonal matrix, whose transpose is its inverse,i.e. $\mathbf{R}^T = \mathbf{R}^{-1}$,and it determinant $det(\mathbf{R}) = 1$. The product of two rotation matrices is still a rotation matrices. On $\Re^2$, the standard rotation matrix has the following form:

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.4}$$

where θ means that a vector has a counterclockwise rotation through angle θ. A given vector $v_0 = [x, y]$ rotated by a counterclockwise angle θ in a fixed coordinate system can be represented by:

$$\mathbf{v}' = \mathbf{R}(\theta)\mathbf{v}_0 \tag{2.5}$$

Which is also shown in the Fig.2.1.

In this thesis, most of the rotation matrices are in 2D Euclidean space.

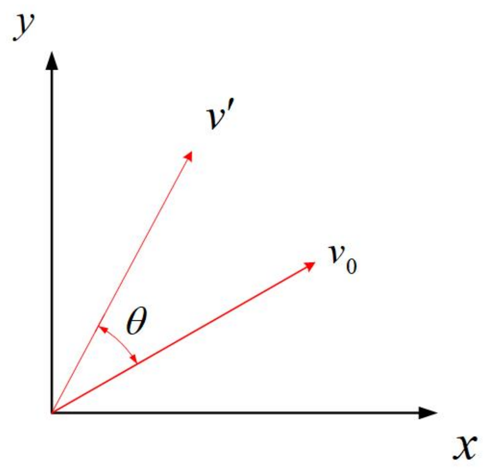


Figure 2.1: Fixed Point Rotation Convention

2.2 Dynamic Model of Vehicles

The dynamical model of vehicles plays a crucial role in the formation control problem. In real world vehicle systems, there are different types of vehicle models, like holonomic model, nonholonomic vehicle model, Euler-Lagrange vehicle model, and high-order dynamic model[15].

In holonomic vehicle systems, the motion of a vehicle in every axis is independent with each other. Thus, the vehicles can directly go to any position without constraints. The general expression of a holonomic vehicle is

$$\dot{p}_i = v_i, \quad \dot{v}_i = u_i, \quad i \in \{1, \ldots, N\} \tag{2.6}$$

Where $p_i, v_i \in \Re^n$,$n \in \{2, 3\}$, are the position and the velocity vectors of vehicles. u_i is the acceleration input of the vehicles.

For nonholonomic models, the model on $\Re^2$ is generally given by

$$\begin{cases} \dot{x}_i = v_i \cos\theta_i \\ \dot{y}_i = v_i \sin\theta_i \\ \dot{\theta}_i = \omega_i, \quad i = 1, \ldots, N \end{cases} \tag{2.7}$$

Where the v_i, θ_i and ω_i are linear velocity, handing direction and angular velocity respectively. The vehicle position in world frame is presented by $p_i = |x_i, y_i|^\top \in \Re^2$. A general illustration of nonholonomic vehicle model is shown in Fig.2.2

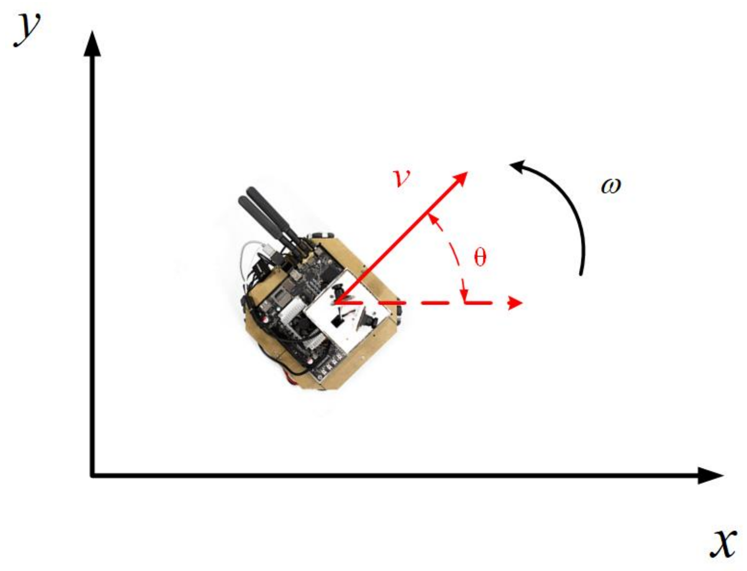


Figure 2.2: Non-holonomic vehicle model

2.3 Graph Theory and Terminology

Basic concepts of graph theory can be found in any graph theory books[16] [17]. In this section, it will introduce the modeling of multi-vehicle systems by using graph theory.

We use the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ to represent a multi-vehicle system with N vehicles,

$1, 2...N$. Where the $\mathscr{V} = \{1, \cdots, N\}$ and $\mathscr{E} = \{(i, j) | j \in N_i, i, j \in \mathscr{V}, i \neq j\}$ represent the vertex and edge sets of graph $\mathscr{G}$ separately. Each vehicle is an element included in the vertex set $\mathscr{V}$. Also, the edge set represent connections among the vehicles. The *connections* can be *direction* or *undirected* connection. For example, if the connections between two vehicles 1 and 2 is undirected, it means both the vehicle 1 and 2 can receive the information from each other. A neighbor set N_i is used to characterizing the connection with vehicle i. The graph set, $\mathscr{G}^f = (\mathscr{V}, \mathscr{E}^f)$, is used to represent desired formation patterns and relation shapes.

2.4 Gradient Formation Control

Gradient formation controllers are based on the gradient descent optimization algorithm, which can find the minimum of a function by using iteration methods. For a general formation control case, consider the formation of **n** vehicles with typical holonomic dynamic 2.6 and a graph pattern $\mathscr{G}$. Between arbitrary two vehicles has an edge $\mathscr{E}_{ij}$ where a corresponding potential function (cost function) V_{ij} can be constructed. When these two vehicles are at the desired relative position, the potential function V_{ij} gets the global minimum value. A typical potential function V_{ij} could have the following conditions[18] .

$$V_{ij} : R^m \to R_{\geq 0} \text{ is continuously differentiable} \tag{2.8}$$

$$V_{ij} = 0 \Longleftrightarrow \|p_i - p_j\| = d_{ij} \tag{2.9}$$

$$\nabla_{p_i} V_{ij} = 0 \Longleftrightarrow \|p_i - p_j\| = d_{ij} \tag{2.10}$$

Where the d_{ij} is the desired relative position between the vehicle **i** and vehicle **j**. ∇_x is defined as $\nabla_{\boldsymbol{x}} \triangleq \left[\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_m}\right]^T$, in 2D case $\nabla_{p_i} = \left[\frac{\partial}{\partial x_i}, \frac{\partial}{\partial y_i}\right]^T$. To calculate all the edges inside the graph, the potential function can be defined as :

$$V = \sum_{(i,j)\in\mathcal{E}(t)} V_{ij} \tag{2.11}$$

Thus, a gradient descent controller for each vehicle can be defined as:

$$u_i = -\nabla_{p_i} \sum_{(i,j)\in\mathcal{E}(t)} V_{ij} \tag{2.12}$$

It has been proved in[19] [20] that the formation will finally converge to the desired pattern $\mathscr{G}^f$. If all the conditions (2.8),(2.9),(2.10) are satisfied.

Sometimes developing the potential functions is a kind of art. Different functions are needed according to various scenarios and tasks.

2.5 Camera Model and Pose Estimation

There are different types of camera models. Take the most common pin-hole cameras for example. A camera model can be divided into intrinsic and extrinsic parts. The intrinsic parameters characterize the pin-hole camera model and lens distortion. The extrinsic matrix includes the translation and rotation of cameras respect to the world frame[21].

The camera model can characterize transformations from a 3D world frame to a 2D image plane. Here using K to represent the intrinsic matrix[22]. Where $[p_x, p_y]$ a translation vector which indicates the offset of 2D points in a image plane.

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}, Where\ \mathbf{K} = \underbrace{\begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}}_{intrinsic\ matrix} \tag{2.13}$$

Where $[x_{pix},\ y_{pix}]^T$ represents a point in camera image planes (pixel coordinates), $[X_c,\ Y_c,\ Z_c]^T$ represents a point in 3D camera coordinates.

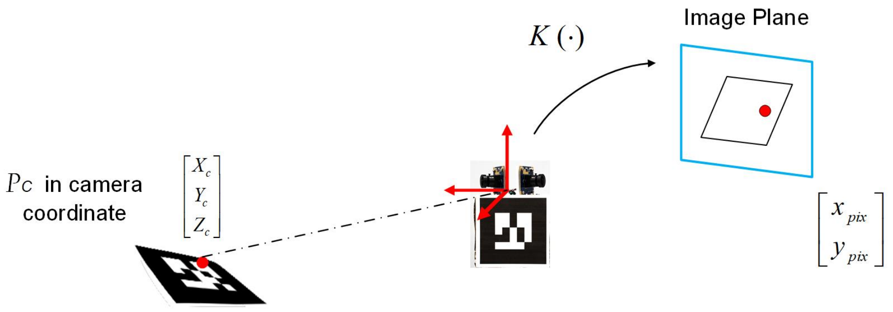

$K(\cdot)$
Image Plane
Pc in camera coordinate
$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$
$\begin{bmatrix} x_{pix} \\ y_{pix} \end{bmatrix}$

Figure 2.3: Camera model transform point from 3D to 2D

Distortion caused by optical components in the camera is another problem in real world image systems. The distortion of an image can be characterized by using a mapping function **D**, $[x_{distorted},\ y_{distorted}]^T = D([x_{ideal},\ y_{ideal}]^T)$:

$$\begin{aligned} x_{\text{distorted}} &= x_{\text{ideal}}\left(1 + k_1^\star r^2 + k_2^\star r^4 + k_3^\star r^6\right) \\ y_{\text{distorted}} &= y_{\text{ideal}}\left(1 + k_1^\star r^2 + k_2^\star r^4 + k_3^\star r^6\right) \end{aligned} \tag{2.14}$$

Where $[x_{distorted},\ y_{distorted}]^T$ represents the point in distorted images, $[x_{ideal},\ y_{ideal}]^T$ means the point in ideal images. $\mathbf{r}$ expresses the distance to the image center, k_1, k_2, k_3 are distortion factors, which respect to different camera and lens.

Extrinsic parameters indicate the pose of cameras related to the world frames, which is represented by the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$. The transformation from world frames $\mathbf{P_w}$ to camera coordinates $\mathbf{P_c}$ is shown in(2.15) and Fig.2.4.

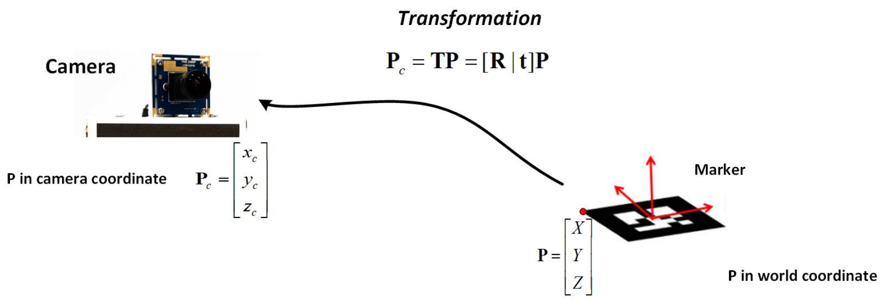

Transformation
$\mathbf{P}_c = \mathbf{TP} = [\mathbf{R} \mid \mathbf{t}]\mathbf{P}$
Camera
P in camera coordinate
$\mathbf{P}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$
Marker
$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$
P in world coordinate

Figure 2.4: The transformation from world frame to camera coordinate

$$\mathbf{P_c} = [\mathbf{R}|\mathbf{t}]\mathbf{P_w} \tag{2.15}$$

Combining the (2.13),(2.14),(2.15), the transformation from world coordinates to image planes can be established in (2.16).

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \mathbf{D}(\mathbf{K} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}) = \mathbf{D}(\mathbf{K}[\mathbf{R}|\mathbf{t}] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}) \tag{2.16}$$

In (2.16), the intrinsic parameters, $\mathbf{D}$, and $\mathbf{K}$ can be identified by doing camera calibration and the $\mathbf{P_w}$ can be known by using prior designed artificial features. Finally, $\mathbf{R}$ and $\mathbf{t}$ can be retrieved By solving (2.16).

Chapter 3

Distributed Formation System Design

This chapter will include the mathematical expression of visual constraint formation problems, the solution to such a problem, and a control system designed to solve these problems.

3.1 Problem Statement

The overall objective of this thesis is to create a distributed formation system working under visual field constraint, with no network information exchange. To express the visual field constraint, a ***relative angle*** γ_{ij} between the vehicle **i** and **j** is introduced based on (2.3).

$$\cos\gamma_{ij} \triangleq f\left(p_i - p_j, \theta_j\right) = \frac{\begin{bmatrix} -\cos\theta_j \\ -\sin\theta_j \end{bmatrix}^T \cdot \begin{bmatrix} x_i - x_j \\ y_i - y_j \end{bmatrix}}{\left\| \begin{matrix} -\cos\theta_j \\ -\sin\theta_j \end{matrix} \right\| \left\| \begin{matrix} x_i - y_j \\ y_i - x_j \end{matrix} \right\|} \tag{3.1}$$

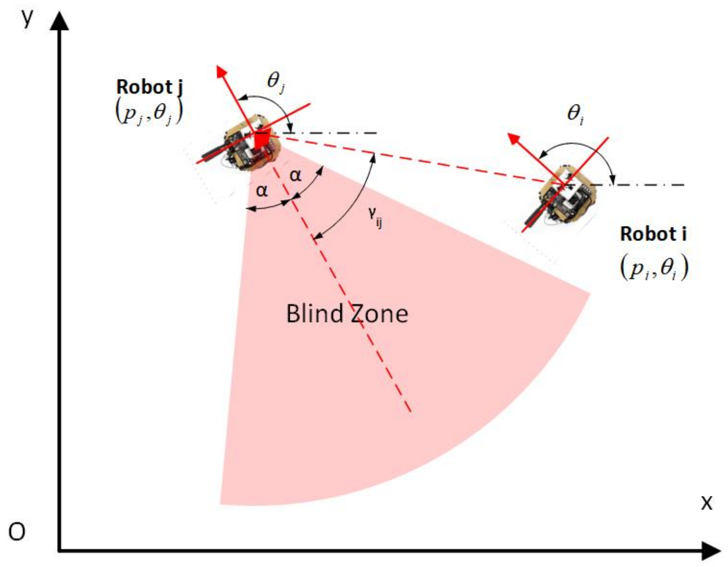


Figure 3.1: Relative angle between vehicle i and vehicle j

Where the $p_i = [x_i, y_j]^T$ represents the position of vehicle i in the world coordinate, θ_j expresses the heading direction of vehicle j. An illustration of relative angle is shown in the Fig.3.1. A blind angle α dividing blind zone equally is imported for the sake of simplicity.

It is obvious that when $\cos\gamma_{ij} < \cos\alpha$, $\gamma_{ij} > \alpha$. It means vehicle i is outside the blind zone of vehicle j. So the neighbor set of vehicle i is expressed by $N_i = \{j \in \mathscr{V} \,|\, \cos(\gamma_{ji}) < \cos(\alpha), \|p_j - p_i\| < R_s, j \neq i\}$. This neighbor set N_i indict the robot j within the visible zone of robot i.

Thus, two objectives are established, which are keeping the visibility among all vehicles and driving the vehicles to a desired formation set. The corresponding expressions are shown:

Objective 1 The restrict $\{\cos\gamma_{ij} < \cos\alpha, \|p_j - p_i\| < R_s, j \neq i\}$ holds for all $i, j \in \mathscr{V}$ during the formation time. The R_s is the maximum visible distance for camera.

Objective 2 The $\lim_{t\to\infty} ||(p_i - p_j) - (h_i - h_j)|| \to 0,\ and\ \theta_i \to \theta_f$ satisfied for $i, j \in \mathscr{G}_f$. Where the $h_i - h_j$ is the desired relative position in world coordinate for vehicle i and j.

3.2 General Solution

Based on the section 2.4 and 3.1, one of an appropriate way to solve the formation control problem is finding a suitable potential function under the given scenario constraint. In[23], a potential function with visibility constraint has been proposed.

3.2.1 Potential Function

The potential function were divided into two parts in the paper[23], V_{ij} and C_{ij}, which is shown in (3.2). The V_{ij} focuses on the formation pattern convergence, and the C_{ij} represents the task of visibility maintenance. k_c and k_v are two weighting factors to balance the C_{ij} and V_{ij}

$$J = \sum_{i=1}^{N} (k_c \sum_{(i,j)\in\mathscr{E}(t)} C_{ij} + k_v \sum_{(i,j)\in\mathscr{E}_{sub}(t)} V_{ij}) \tag{3.2}$$

Where $\mathscr{E}_{sub}(t)$ means a sub edge set, which vehicle i and j inside it is in a visible zone within the distance **r** . **r** is a design factor given by $\frac{R_s - r}{R_s} < 1$ and $Rs > r >$

$max\{||h_i - h_j||\}$. The expression of $\mathscr{E}_{sub}(t)$ are shown in below.

$$1.\mathscr{E}_{sub}(0) = \mathscr{E}(0) \cap \mathscr{E}^f$$
$$2.\mathscr{E}_{sub}(t_+) = \mathscr{E}_{sub}(t_-) \cup \{(i,j)| \quad \|p_i(t) - p_j(t)\| \leq r \ , \cos(\gamma_{ji}(t)) < \cos(\alpha), i, j \in \mathscr{V})\} \tag{3.3}$$

The cost function V_{ij} is presented

$$V_{ij} = \begin{cases} \frac{\|(p_i - p_j) - (h_i - h_j)\|^2}{\|p_i - p_j\|^2}, \|p_i - p_j\| \in (0, \|h_i - h_j\|] \\ \frac{\|(p_i - p_j) - (h_i - h_j)\|^2}{(R_s - \|p_i - p_j\|)^2}, \|p_i - p_j\| \in (\|h_i - h_j\|, R_s) \end{cases} \tag{3.4}$$

It is obvious that the equation(3.4) satisfies the requirements, (2.9), and (2.10). More over it also has $\lim\limits_{\|p_i - p_j\| \to 0} V_{ij}(p_i - p_j) = +\infty$ and $\lim\limits_{\|p_i - p_j\| \to R_s} V_{ij}(p_i - p_j) = +\infty$, which prevent the vehicle i collision with vehicle j or excess visibility distance.

The cost function C_{ij} is represented by $C_{ij} = \rho(\|p_i - p_j\|) J_v(\gamma_{ji})$, where $\rho(\|p_i - p_j\|)$ is a smooth function to avoid effects of $J_v(\gamma_{ji})$, when a sudden change of $\mathscr{G}(t)$ happen. $J_v(\gamma_{ji})$ is used to drive vehicles goes out the blind zone(3.5).

$$J_v(s) = \begin{cases} 0, & s > \beta \\ \int_\beta^s \frac{\cos_\beta - \cos(\tau)}{\cos(\alpha) - \cos(\tau))^3} d\tau, & s \in (\alpha, \beta] \\ \int_\beta^s \infty, & s \in [0, \alpha] \end{cases} \tag{3.5}$$

In the (3.5), a transition zone is defined by using α and β. When γ_{ij} is in the transition zone $(\alpha, \beta]$, the $J_v(\gamma_{ij})$ is monotonically decreasing.

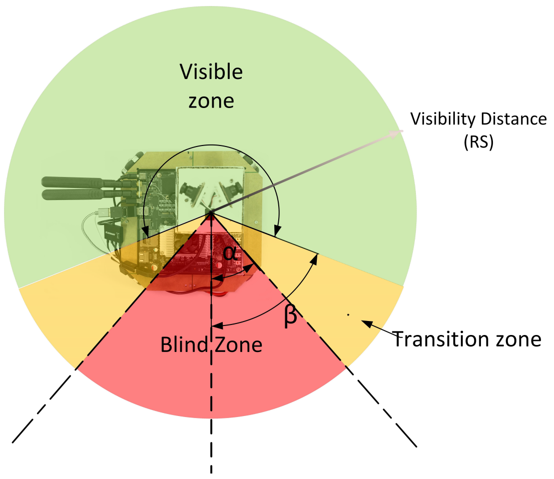


Figure 3.2: Transition zone used in $J_v(s)$

The $\rho(\|p_i - p_j\|)$ is given by

$$\rho\left(\|p_i - p_j\|\right) = \begin{cases} 1, & \|p_i - p_j\| \in [0, r) \\ \frac{1}{2}\left(1 + \cos\left(\pi \frac{\|p_i - p_j\| - r}{R_s - r}\right)\right), & \|p_i - p_j\| \in [r, R_s] \\ 0, & \|p_i - p_j\| \in [R_s, \infty) \end{cases} \tag{3.6}$$

3.2.2 Local Controller

The controller designed by (2.12) can only be directly used in the vehicle having holonomic kinematic model. Thus, a method convert the gradient to linear and angular velocity need to be proposed[23].

$$\begin{aligned} v_i &= -\operatorname{sgn}\left(\frac{\partial J}{\partial x_i}\cos(\theta_i) + \frac{\partial J}{\partial y_i}\sin(\theta_i)\right)\|\nabla_i J\| + v_0 \\ \omega_i &= \left(\frac{\partial J}{\partial x_i}\sin\left(\frac{\theta_i + \theta_f}{2}\right) - \frac{\partial J}{\partial y_i}\cos\left(\frac{\theta_i + \theta_f}{2}\right)\right) v_0 \\ &\quad - \sin\left(\frac{\theta_i - \theta_f}{2}\right) \end{aligned} \tag{3.7}$$

Where $\nabla_i J = [\frac{\partial J}{\partial x_i}, \frac{\partial J}{\partial y_i}]^T$. And sgn(s) is a signum function defined as:

$$\operatorname{sgn}(x) := \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{3.8}$$

The convergence ability of the proposed local controller in (3.7) is proved in[24] by Lyapunov method and a detail derivation is given in the page57 Appendix C. It is obviously that

$$\frac{\partial J}{\partial x_i} = \frac{\partial J}{\partial y_i} = 0 \quad \rightarrow \quad v_i = v_0 \tag{3.9}$$

And when $\frac{\partial J}{\partial x_i} = \frac{\partial J}{\partial y_i} = 0$ & $\theta_i = \theta_f$ the $\omega_i = \dot{\theta}_i = 0$.

3.3 System Design

3.3.1 Distributed Multi-vehicle System

Due to the communication is restricted, and each vehicle is seen as an individual unit. Thus, a distributed system are constructed to verify the control algorithm. In a distributed environment each vehicle perceives environments and makes decisions independently. All the setup related with the experiments are predefined inside in each vehicle include relative positions, desired velocity, handing, etc.

A general illustration of the multi-vehicles formation system is shown in the 3.3. With N vehicles included, the system can be established to test the effectiveness of convergence and visual constraint of the proposed formation control algorithm.

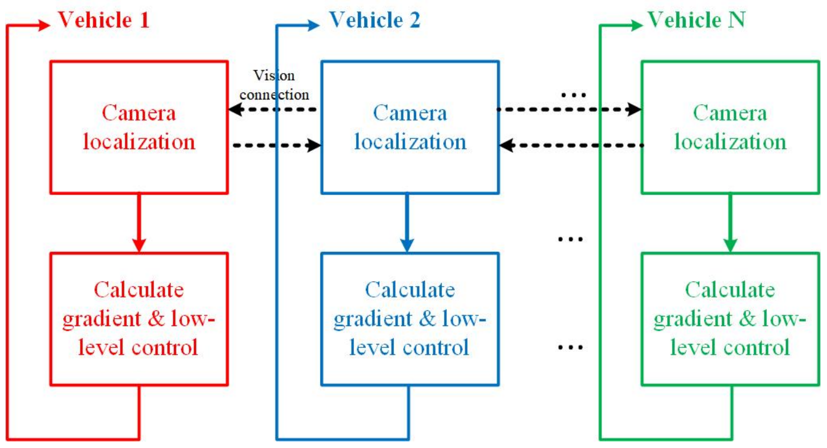


Figure 3.3: Multi-vehicle Formation System with Visual Constraint

3.3.2 Embedded System Design

Based on (3.7), a local controller can be designed to maintain the formation pattern which only relay on the information explored by itself. Thus, we can focus on design control for each vehicle individually.

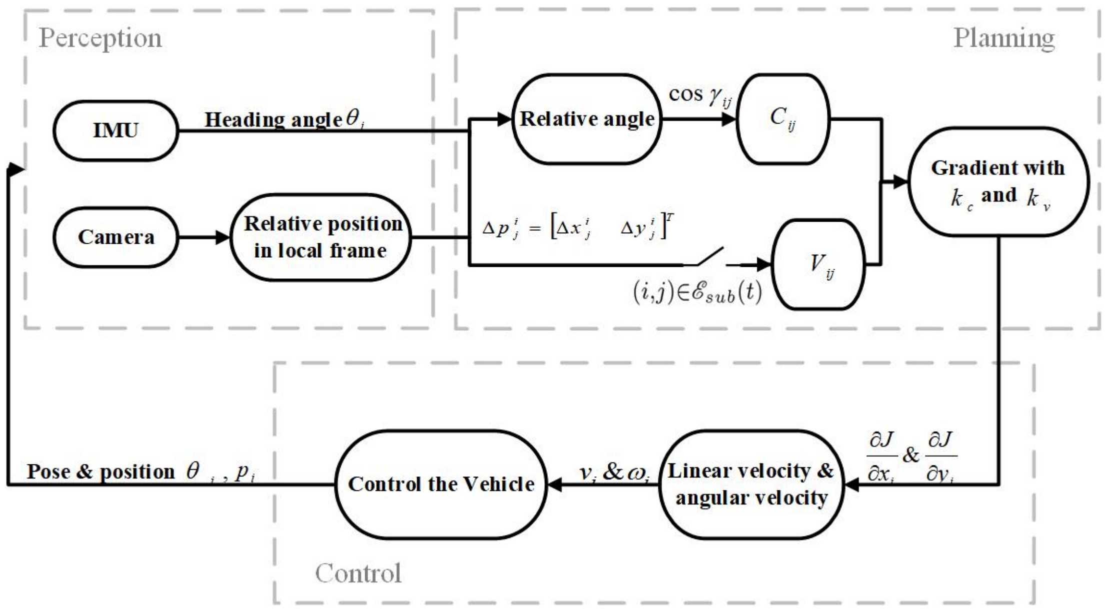

Perception
Planning
Control
IMU
Heading angle θ_i
Camera
Relative position in local frame
Relative angle
$\cos\gamma_{ij}$
C_{ij}
$\Delta p^i_j = [\Delta x^i_j \quad \Delta y^i_j]^T$
V_{ij}
$(i,j)\in\mathscr{E}_{sub}(t)$
Gradient with k_c and k_v
$\frac{\partial J}{\partial x_i}$ & $\frac{\partial J}{\partial y_i}$
Linear velocity & angular velocity
v_i & ω_i
Control the Vehicle
Pose & position θ_i, p_i

Figure 3.4: The control structure for each vehicle

Combining the previous results with a real word sensing environment scenario, including the information from cameras and IMUs(Inertial Measurement Unit), an illustration of control structure is shown in Fig3.4 . The whole structure can be divided into three parts (1) Percetion (2)Planning (3)Control. In the percetion section,the IMUs are used to get the heading direction of themselves, and cameras are used to resolve the relative pose between vehicles. In the planning part, based on the pre-processing information from last step, the $\frac{\partial J}{\partial x_i}$ and the $\frac{\partial J}{\partial y_i}$ are are calculated. In the flow chart the C_{ij} is given by (3.5)and(3.6), V_{ij} is given by (3.4) respectively. As long as the robot j is detected by the robot i, we calculate C_{ij} and only when $(i, j) \in \mathscr{E}_{sub}(t)$ the V_{ij} is included in the cost function. The control part is responded for regulation the speed on each motor based on given linear and angular velocity.

Chapter 4

Implementation Detail

This chapter will introduce the setup of experiment equipment and the development of software.

4.1 Overview of Vehicle Setup

It is a comprehensive work to design an embedded system including variety of different architecture, such as software and hardware, which need cooperation to get decent performance. According to needs of visual formation, the primary task of the project is setting a platform with appropriate hardware and open-source software that can be used to test formation algorithm in indoor environments. Priority is give to the object recognition and pose estimation. It is essential to the existing formation framework in section 3.3 for on-board autonomous control .

Fig.4.1 is a general architecture that briefly illustrate the main components used in this project. The main program runs on NVIDIA TX2 with Linux system. Robot Operation System[25] (ROS) are used to fulfill high-level action which finally send the desired velocity to the MCU (Micro Controller Unit). MCU is responsible for low-level motors control. The detailed features will be talked in the following sections.

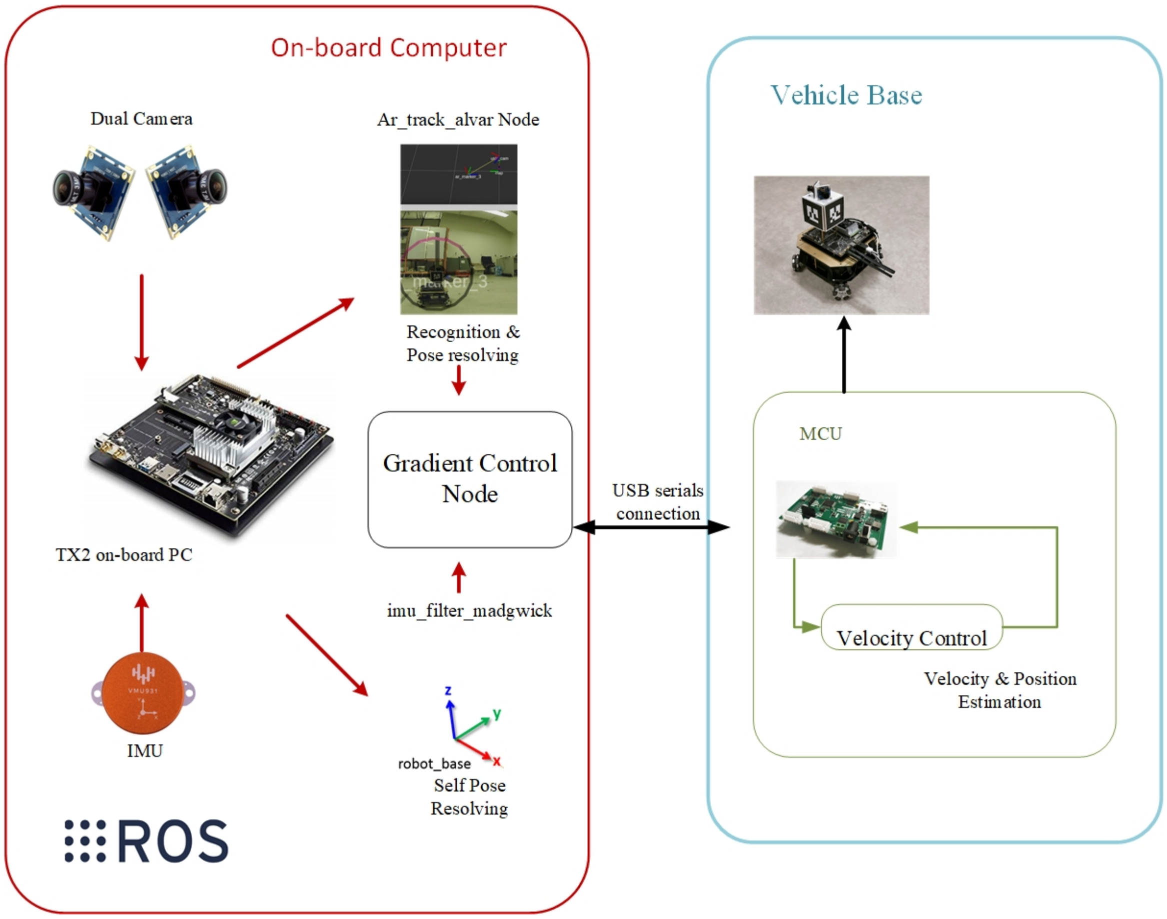


Figure 4.1: Project Architecture

4.2 Configurations of the Vehicle

Each vehicle is seen as one basic unit for the formation system. It is necessary to consider the kinematics and some on-board sensor performance of the vehicle.

4.2.1 Vehicle Kinematics and Control

Nowadays, there are several of different kinematic models for vehicles, such as, differential two wheels, omnidirectional, ackerman, mecanum, etc. In this experiment, four omnidirectional wheel vehicles are used to test the algorithm. A general sketch of the omnidirectional wheel vehicles is shown in Fig.4.2. With the $\phi = 45°$ in graph.4.2, the wheels are orthogonal. The forward and inverse kinematic models[26] of the omnidirectional vehicles are shown below (4.1),(4.2):

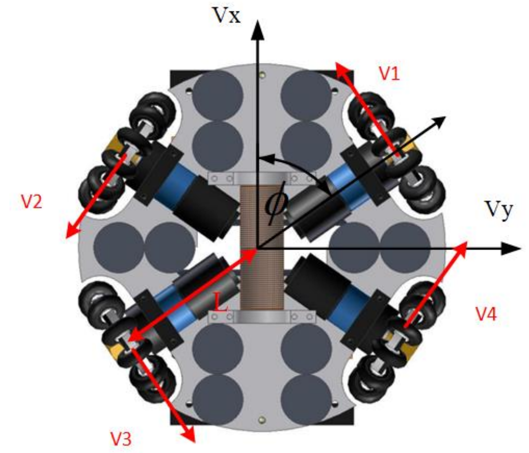


Figure 4.2: The Sketch of Omnidirection Wheel Vehicle

$$\mathbf{V}_r = \begin{Bmatrix} v_x \\ v_y \\ \omega \end{Bmatrix} = \frac{1}{4} \begin{bmatrix} -\sqrt{2} & -\sqrt{2} & \sqrt{2} & \sqrt{2} \\ \sqrt{2} & -\sqrt{2} & -\sqrt{2} & \sqrt{2} \\ \frac{1}{L} & \frac{1}{L} & \frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{Bmatrix} \tag{4.1}$$

$$\mathbf{V}_w = \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{Bmatrix} = \begin{bmatrix} \frac{-\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & L \\ \frac{-\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & L \\ \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & L \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & L \end{bmatrix} \begin{Bmatrix} v_x \\ v_y \\ \omega \end{Bmatrix} \tag{4.2}$$

Where $\mathbf{V}_w = [v_1\ v_2\ v_3\ v_4]^T$, $v_1\ v_2\ v_3\ v_4$ are the wheel velocities in the active direction. $\mathbf{V}_r = [v_x\ v_y\ \omega]^T$,$v_x$ and v_y are the translational velocities of the robot center, ω is the angular velocity about the robot center L is the distance from the robot center to steering axis. The omnidirectional vehicles is a kind of holonomic system. But in this thesis, we only use the v_x as linear velocity and ω as angluar velocity to control the vehicle. There is **no** v_y included in high-level control part.

For the velocity control of chassis, four PID controllers are used to track the desired linear and angular velocities by controlling four motor drives. An illustration of the low-level velocity control is shown in 4.3.

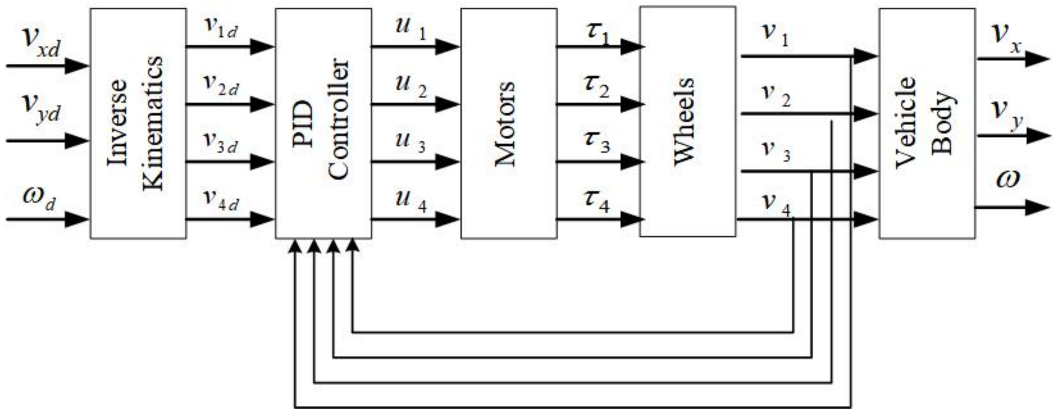


Figure 4.3: PID Control of Vehicle Base Chassis

4.2.2 Vehicle Specifications

The hardware structure of vehicles are shown in Fig.4.4. The control units of vehicles are divided to two parts: (a) Master Controller used as a High-level controller (b) Servo controller used as a low-level controller. The low-level controller is a customized board based on STM32f103RCT6, which is used to drive the motors according to the linear and angular velocities sent from high-level controller, meanwhile, resolve the real-time linear and angular velocity based on encoder data. The NVIDIA TX2 developer board is used as the High-level controller. All computing power consuming programs are calculated on TX2, like image processing and motion planning.The NVIDIA TX2 controls the vehicle just by sending linear and angular velocity to the low-level controller. Meanwhile, the peripheral sensors are connected to the TX2, including two cameras and IMU. The laptop is used to control all vehicles start/stop or trigger special event, like changing heading direction.

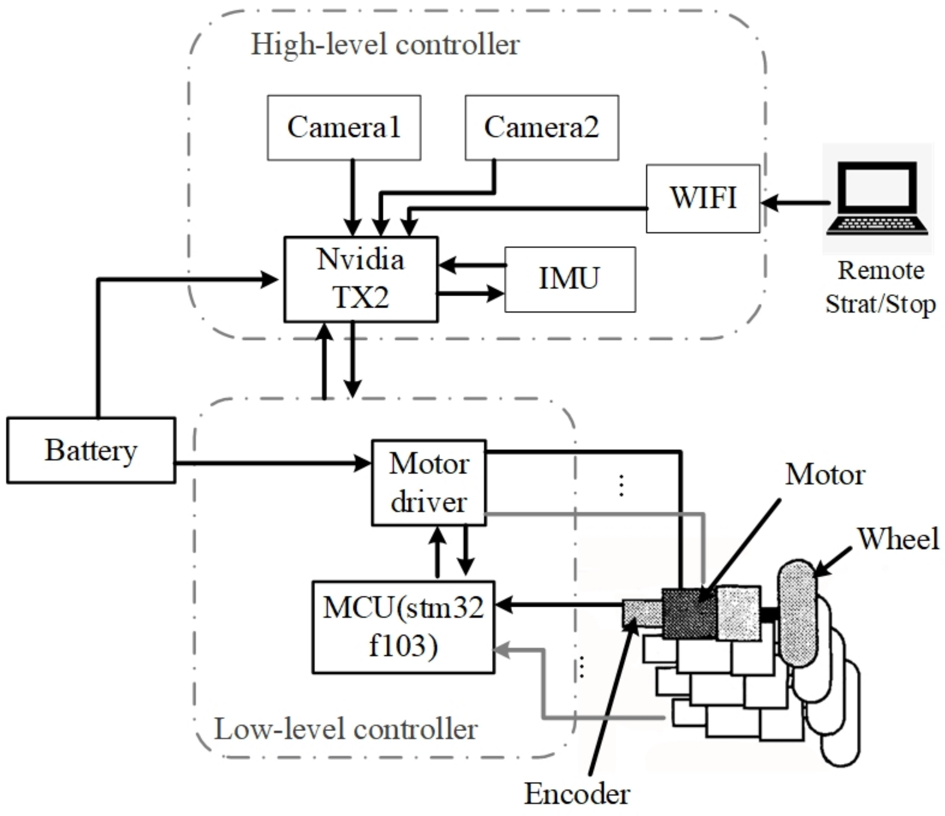


Figure 4.4: The Hardware Structure of Vehicle

A picture of the omnidirectional vehicles used in this thesis is designed and constructed as illustrated in Fig.4.5. Each side of the vehicle is 260mm, the height of the vehicle is 340mm, and the motor drives and controller are placed in the space between the platform and ground. With the selected motor and arrangement mechanism, the vehicles have maximum linear and angular velocity 1.2 m/s and 5.3 rad/s respectively. For detail about the hardware features, it is shown in the page52 Appendix A.

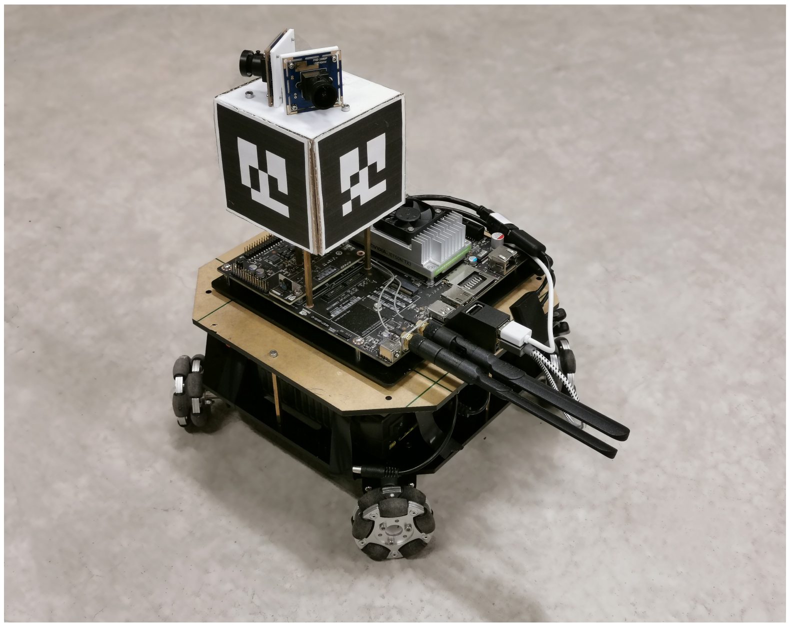


Figure 4.5: The photo of vehicle

4.2.3 Localization System

Localization system here means both the vision capture method and the features used for recognition. This vehicle equipped a marker box with four markers mounted on the top, which is used as the artificial features to identify different vehicles and do the localization as shown in Fig.4.5. By using four orthogonal markers, the vehicle can be detected in any direction An illustration can been found in Fig.4.6.

Each marker forms 90 degree angles with two adjacent markers. Also, the markers on every surface are unique for the convenience of identification the nubmer and heading of the vehicles. As shown in Fig4.6, taking vehicle 2 for example, the marker 5 is on the back of the marker box. When other vehicles get the relative handing of marker 5, it can resolve the relative handing between itself and vehicle 2 by adding 180°.

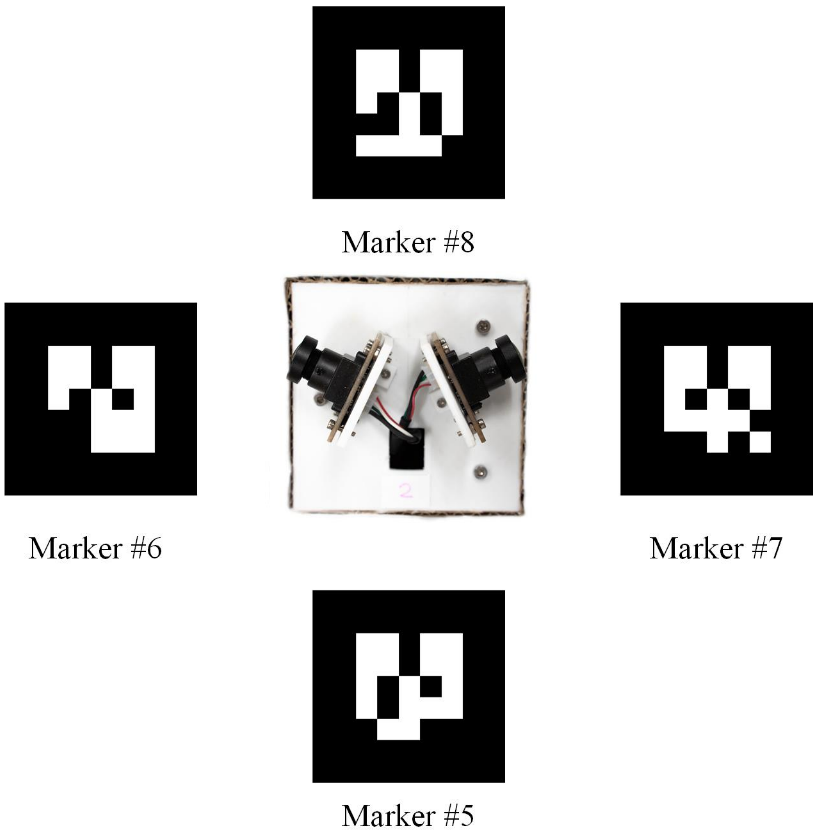


Figure 4.6: The Marker Box Illustration

To mimic a super wide angle fish-eye camera, two cameras are mounted in opposite direction with a 170° horizontally angle of view and the rotation angle $\theta_l = \theta_r = 60°$ which leaves an angle of 70° for blind zone as shown in Fig.4.7. In Fig.4.7, the green areas represent visible zone, the red represent the blind zone. The resolution for each camera is full HD 1920*1080 and support 30 fps at 1920*1080 mode. A demonstration of the photo captured by two cameras mounted on the vehicle is shown in Fig.4.8. The green areas in Fig.4.8 represent the overlap area captured in both two cameras simultaneously. The angle of these green areas is around 45° in reality.

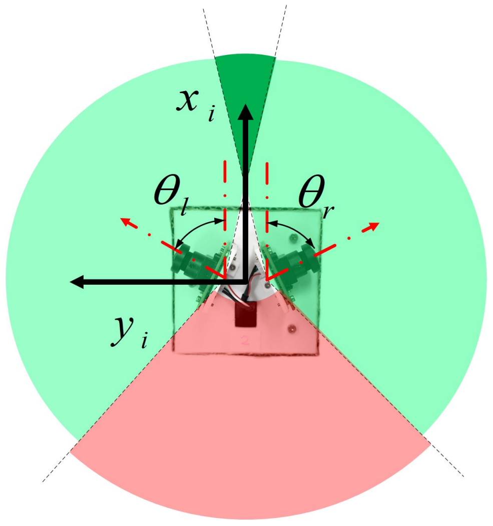


Figure 4.7: The Mount of Cameras

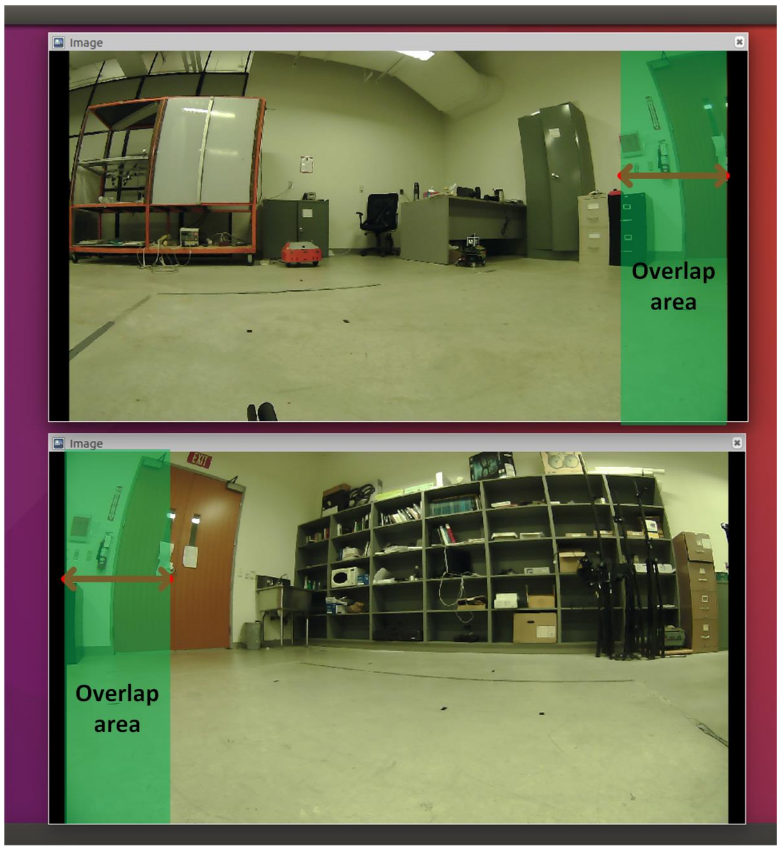


Figure 4.8: The Photos Captured from Two Cameras

4.3 Software Environments

This section will introduce the software environments and the packages or libraries used in the thesis will be introduced.

4.3.1 ROS

ROS is the abbreviation of **R**obot **O**perating **S**ystem and ROS is a distributed framework for developing and testing robotics software or algorithms[27]. It provides bunch of services, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. ROS has a distributed developing and running structure. By Using the publish-subscribe mechanism, each node (program or process) can subscribe to so-called topics and push message to others. ROS also provide the data recording package named rosbag. By using rosbag, it is easy to record different types message data for late analyzing like, image and odometer. Hence, the control of autonomous vehicle is mainly developed and ran on ROS.

4.3.2 `Ar_track_alvar`

`Ar_track_alvar` is a package in ROS based on ***ALVAR*** library. ***ALVAR*** is a suite of SDKs and products that help researchers and engineers to create augmented reality applications[28]. In `Ar_track_alvar`, it provides the fast object recognition and tracking through fiducials. The fiducials or markers can be also generated with `Ar_track_alvar`, which looks like QR-code. The markers with several features help to be detected, such as clear border and unique pattern. Combined with the prior knowledge, the size of markers and camera intrinsic matrix, the 2D image plane will be transformed to 3D world frame. Thus, a 6 DoF pose estimation can be realized in the end. A demo of using the `Ar_track_alvar` package to track vehicles is shown in

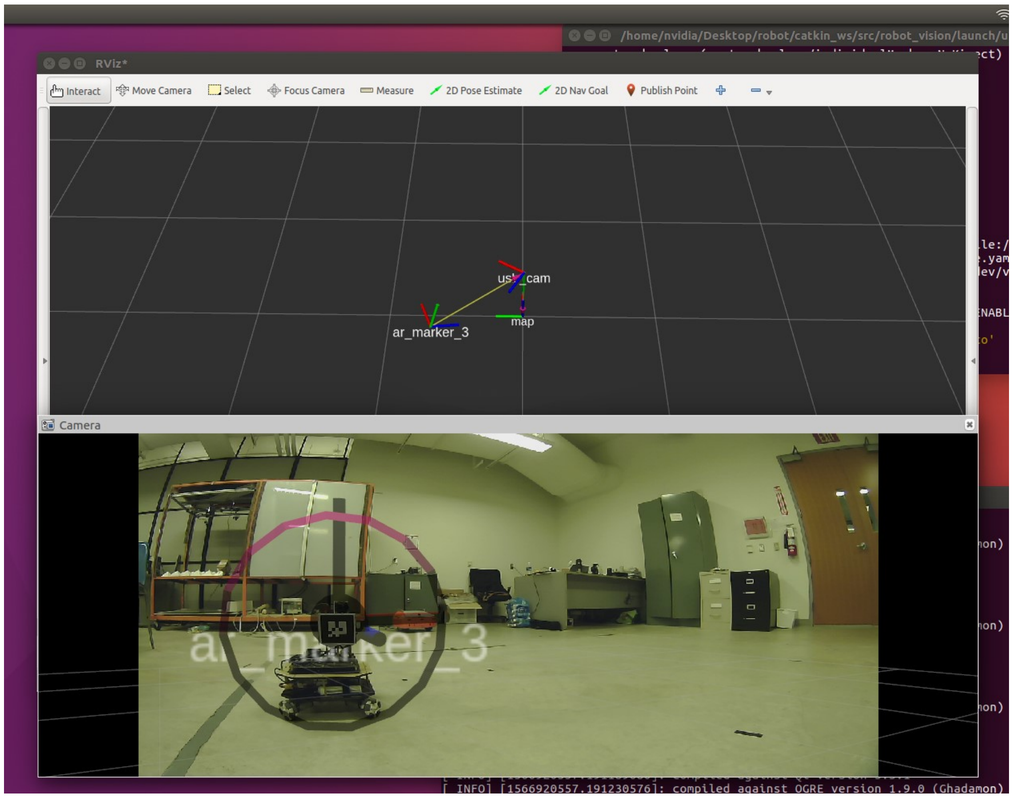


Figure 4.9: A Visualization Demo of `Ar_track_alvar`

Fig.4.9

4.3.3 Camera Calibrator

`Camera_calibrator` is a camera calibration tool in ROS based on OpenCV camera calibration library[29]. `Camera_calibrator` allows easy calibration of monocular or stereo cameras using a checkerboard calibration target. A demo of using checkerboard to calibration is show in Fig.4.10.

4.3.4 MATLAB

Matlab(matrix laboratory) is one of the most widely used numerical computing environment. In this thesis, Matlab is used for reading and analyzing the data recorded by */odom* message. All the graphs in Chapter5 are plotted by Matlab.

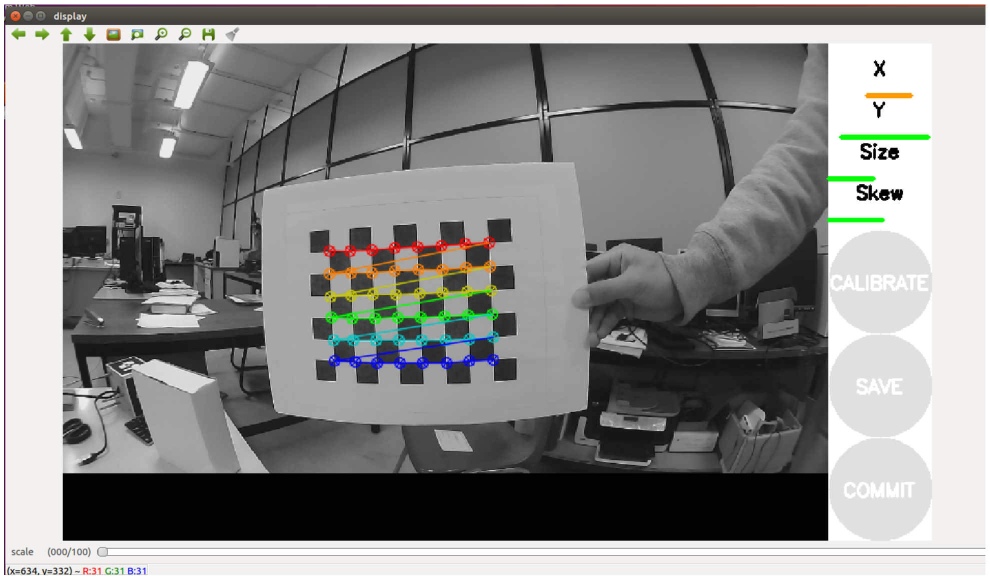


Figure 4.10: Checkerboard Borad for Calibration

4.3.5 Other Libraries and Packages

Some libraries and packages used in the project are briefly introduced in the following.

Gscam[30] based on gstreamer library is used as camera driver in the project. ***Rosbag***[31] is a set of tools for recording from and playing back to ROS topics. A remote control node, ***udp_server***, is programmed with universal UPD protocol to receive command from the laptop. ***Imu_complementary_filter***[32] is imported to fuses angular velocities, accelerations, and (optionally) magnetic readings from a generic IMU device into a quaternion. ***Rosserial***[33] is a set of library used to communication between embedded system devices and ROS system devices.

Chapter 5

Experiments and Result Analysis

In this chapter, the procedure of experiment will be introduced and results will also be analyzed. In order to test the performance and effectiveness, some different initial poses, heading directions, and desired velocity conditions are configured during the experiments. The stability, visual constraint and the synchronization of linear and angular velocity will be evaluated as the system performance.

5.1 Indoor Environment

All the experiments are finished in the indoor environment with no global localization information can be used. In Graduate Control and Robotics Lab shown in Fig5.1, it shows the environment to test the convergence ability and another long range turning experiment for testing visual constraint using the aisle on the 1st floor in Center for Engineer Innovation (CEI).

A general setup can be seen in the Fig.5.2. With the number of N vehicles in the system, each vehicle is set to run independently. By using the UDP protocol, the laptop computer is only used to send command to start, end the experiment and trigger to change some target objectives like velocity, formation pattern and handing

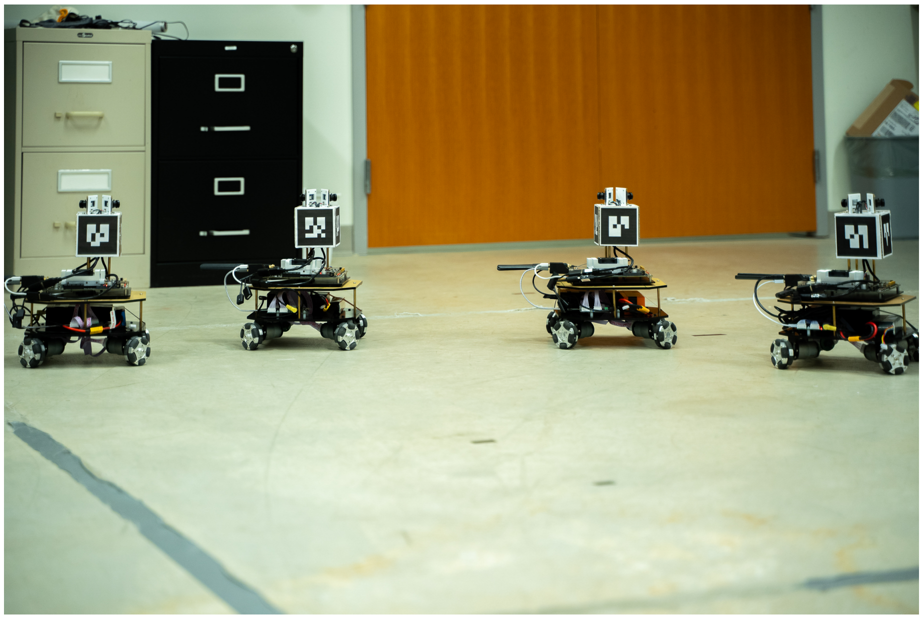


Figure 5.1: The indoor environment with 4 vehicles

direction.

To test the system, a set of random initial poses are chosen to only test the convergence and a set of neat initial patterns are chosen to analysis several kinds of performance. However, the initial pose for $\mathscr{G}(0)$ should be connected.

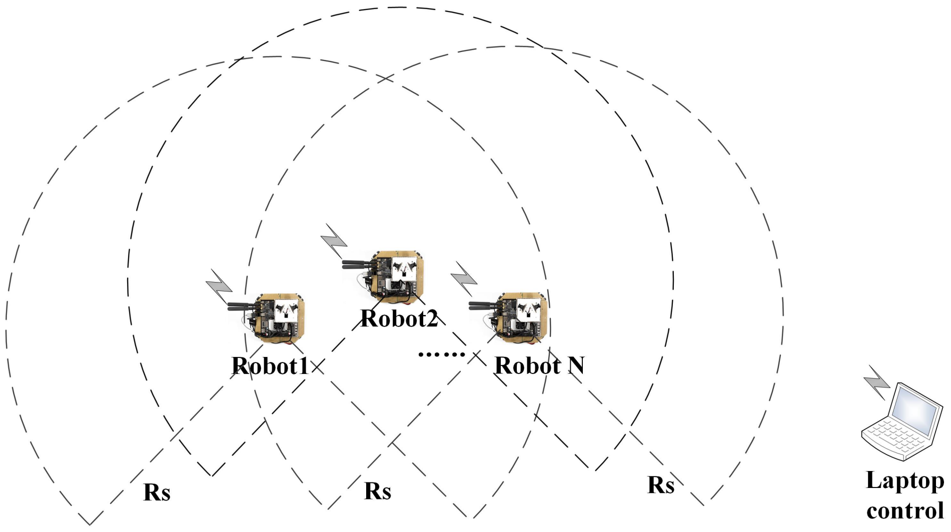


Figure 5.2: A General Setup of the Formation System

The desired formation pattern in these two experiments are setting to be a parallelogram shape, which satisfy the visual constraint. A sketch of the desired formation pattern is shown in the Fig.5.3. The relative position for two experiments in world frame are set to $h_1 = [0\ 0]^T$, $h_2 = [0.7\ 0.4]^T$,$h_3 = [1.4\ 0]^T$,$h_4 = [2.1\ 0.4]^T$.

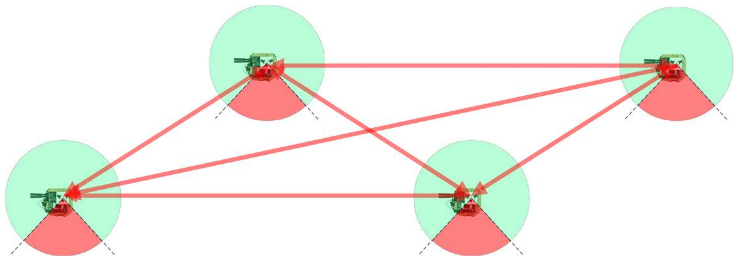


Figure 5.3: A Predefined Formation Pattern Target

5.2 Random initial position

In order to test the convergence ability, several sets of initial positions are chosen which can be seen in Fig.5.4. Some parameter used in the experiments are listed in the Table5.1. The video containing whole process of three different initial poses is upload on the following website address.

https://youtu.be/Tp5egQl2Bg0.

In Fig.5.4, it also shows some screen shots in this video with corresponding time. In Fig.5.4(a), Fig.5.4(b) and Fig.5.4(c), the experiments are using 3 different inital poses.Form the video and Fig.5.4, it can be notice that, in the most of case, the formation can converge to the desired pattern within 10 seconds. And a path record for the straight line initial position(Fig.5.4(c)) is shown in Fig.5.5

α(factor indicate the bland zone)	35°
k_c(weight factor of C_{ij})	0.12
k_v(weight factor of V_{ij})	0.2
ϵ_v(boundary of transition zone)	0.5
R_s(maximum visual distance)	5m
r(design factor of range)	3m
θ_f(desired heading angle)	90°
v_0(desired velocity)	$0.2m/s$

Table 5.1: Parameters Used in the Experiment

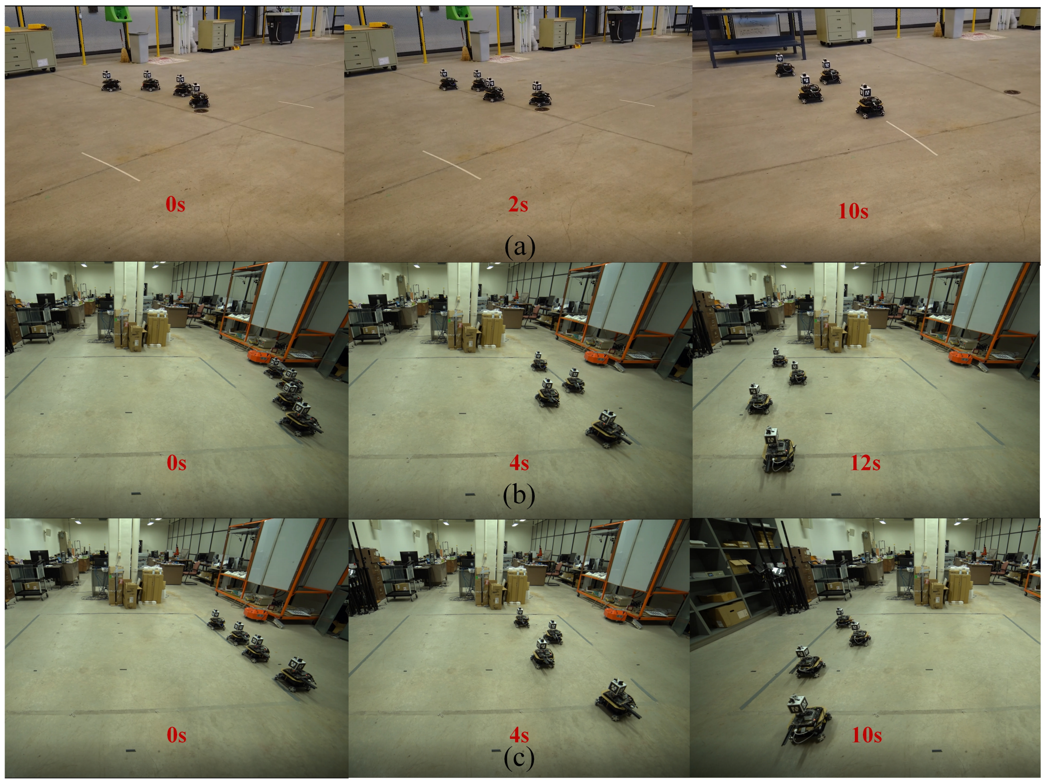


Figure 5.4: Experiment of Different Initial Poses

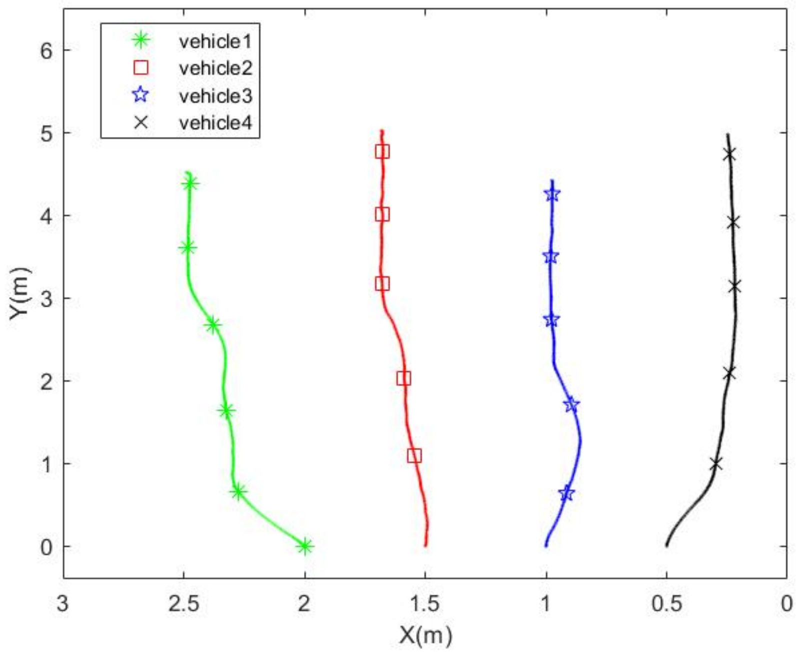


Figure 5.5: Path of the Experiment in Fig5.4.(c)

5.3 Long Range Turning Test

In this experiment, the vehicles will change to different heading direction to test the visual constraint and the influence under disturbance of changing objective target. Considering the convenience of analysis the performance, a set of straight line initial position $z_1 = [0\ 0\ 90°]^T$,$z_2 = [0.5\ 0\ 90°]^T$,$z_3 = [1\ 0\ 90°]^T$,$z_4 = [1.5\ 0\ 90°]^T$ is chosen. In the experiment, vehicles begin with an objective heading direction $\theta_{f1} = 90°$ first. After all robots converged into the desired formation and direction, we change the objective handing direction to $\theta_{f2} = 135°$. All parameters used in the experiment are listed in the Table.5.2.

The full experiment process can be seen at

https://youtu.be/5x1tOIw7TJc

Some selected screenshots are listed in Fig.5.6

α(factor indicate the bland zone)	35°
k_c(weight factor of C_{ij})	0.08
k_v(weight factor of V_{ij})	0.12
ϵ_v(boundary of transition zone)	0.5
R_s(maximum visual distance)	5m
r(design factor of range)	3m
θ_{f1}(heading before turning)	90°
θ_{f2}(heading after turning)	135°
v_0(desired velocity)	$0.2m/s$

Table 5.2: Parameters Used in the Experiment

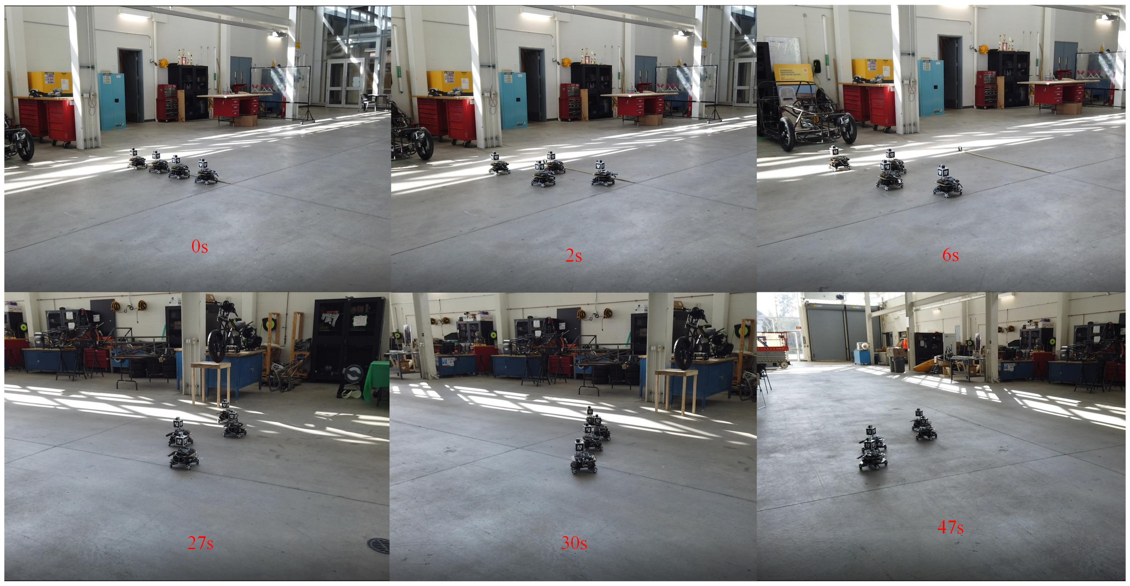


Figure 5.6: Experiment of formation turning test

5.3.1 Result Analysis

The result are retrieved from the odometer of each vehicle separately. By adding the initial offset, the trajectories of all the vehicles are collected together in Fig.5.7. And the orientations and velocities of four vehicles are illustrated in Fig.5.8 and Fig.5.9 separately.

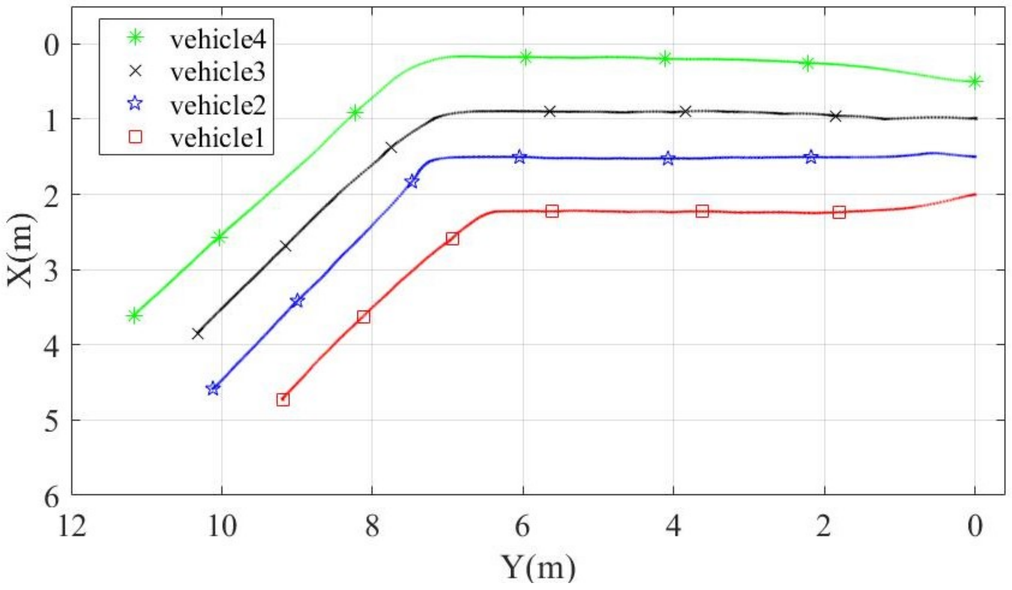


Figure 5.7: Trajectories of all the Vehicles

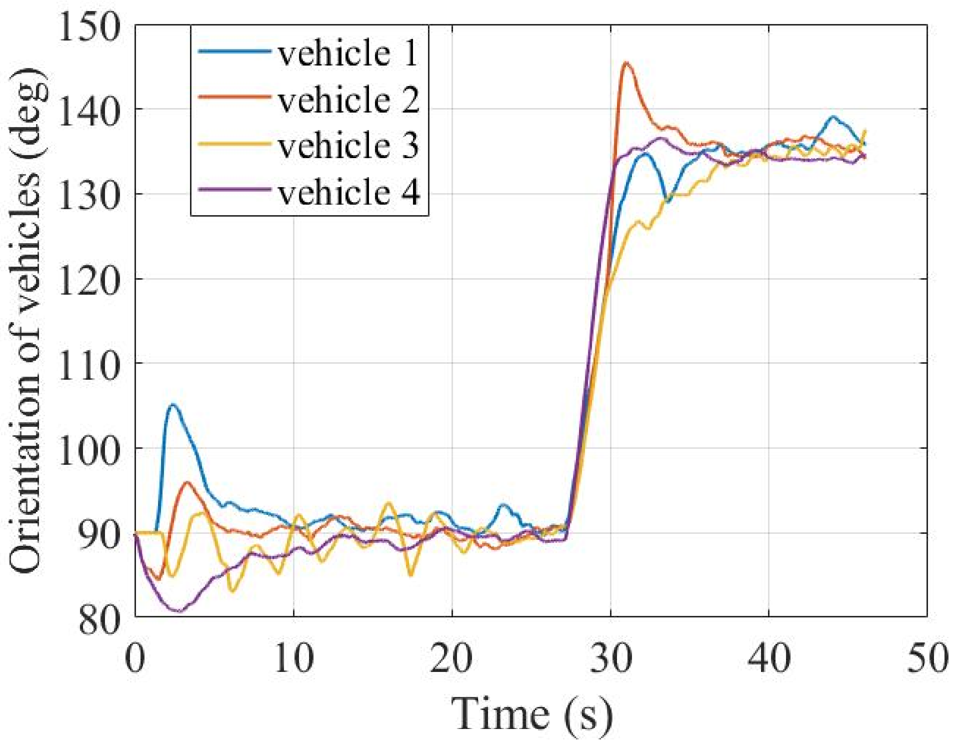


Figure 5.8: Orientation of vehicles during the experiment

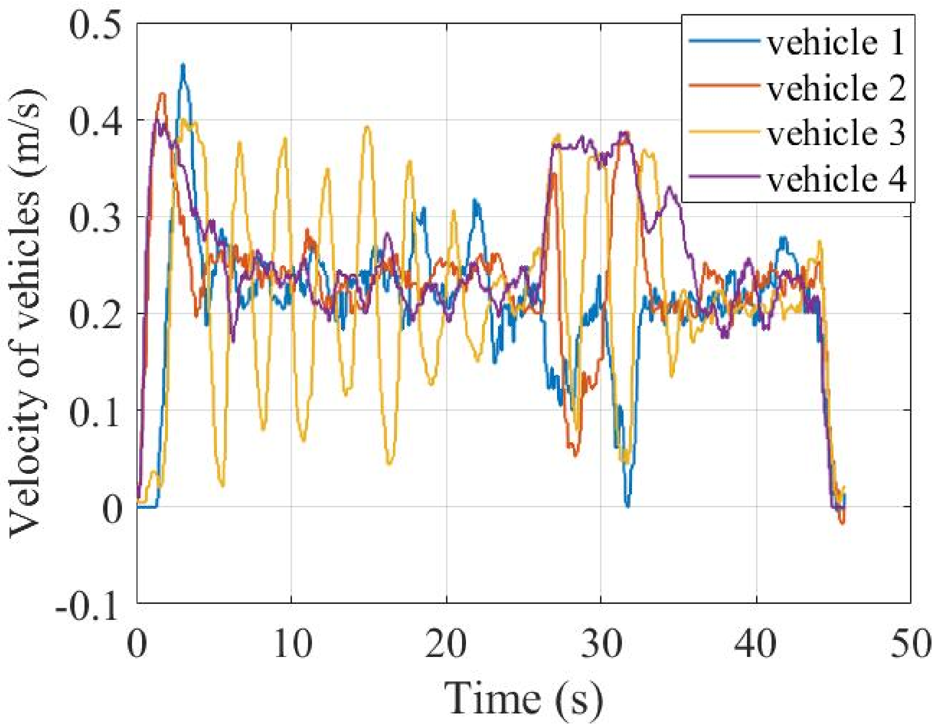
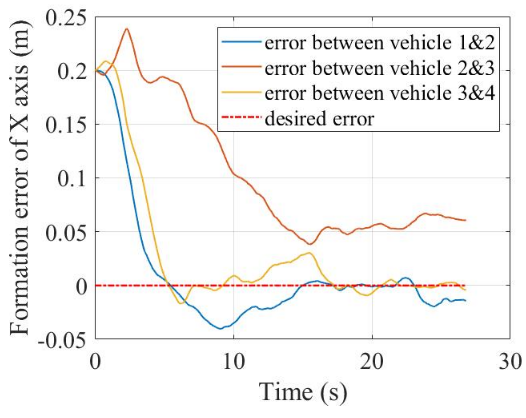


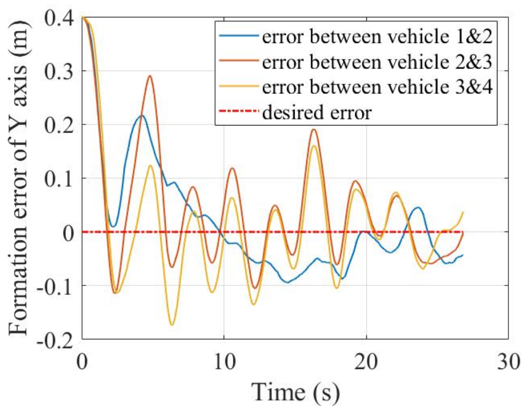
Figure 5.9: Velocity of vehicles during the experiment

From the Fig.5.8 and Fig.5.9, it can be noticed that the orientation angles of all vehicles fluctuate around the desired handing $\theta_{f1} = 90°$ before turning, and around the $\theta_{f2} = 135°$ after changing desired heading direction. And the velocity of all vehicles fluctuate around the desired velocity $v_0 = 0.2m/s$ both before and after changing the objective handing angle.

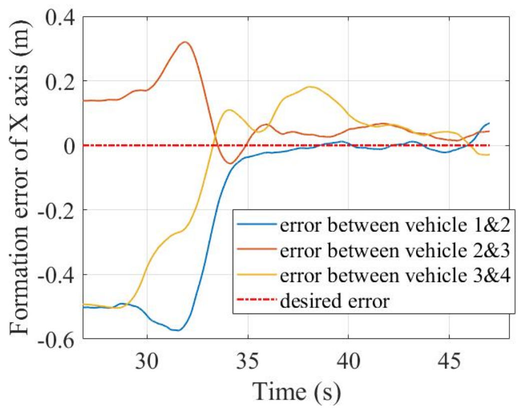
The Formation errors corresponding with the desired pattern are shown in the Fig.5.10. Fig.5.10(a),(b) are the formation errors of X and Y axis with world coordination before the turning action, Fig.5.10(c),(d) are the errors after turning action.



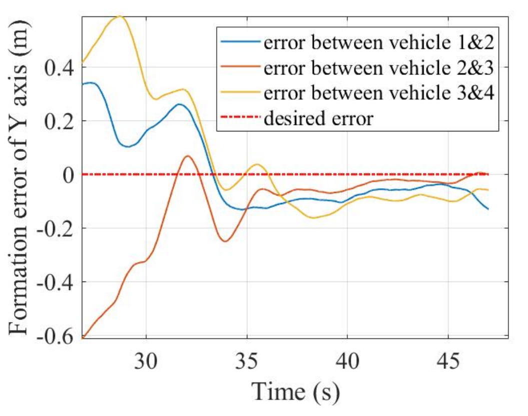
(a)



(b)



(c)



(d)

Figure 5.10: (a)(b) Formation Error of X and Y Directions before Turning, (c)(d) Formation Error of X and Y Directions after Turning

From table5.3 to 5.6, it collects all the mean absolute errors of relative positions and orientations in steady states both before and after the turning action.

In Fig.5.11, the relative angles γ_{ij} are illustrated in the view of vehicle 1 and 3. All relative angles are larger than $35°$ during the whole formation procedures which means no vehicle falling into other vehicles blind zone during the experiment.

Mean Absolute Error

Vehicle# / Position[m]	1&2	2&3	3&4
X	0.0126	0.0684	0.0077
Y	0.0429	0.0552	0.0530

Table 5.3: MAE of relative positions before turning.

Vehicle# / Orientation[°]	1	2	3	4
Yaw	1.1758	0.7120	1.5396	1.0856

Table 5.4: MAE of orientations before turning

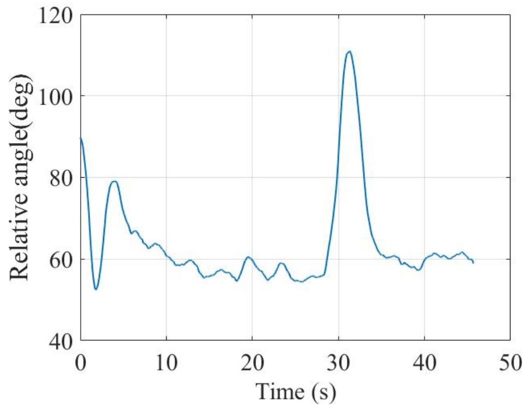
Mean Absolute Error

Vehicle# / Position[m]	1&2	2&3	3&4
X	0.0137	0.0418	0.0456
Y	0.0622	0.0246	0.0863

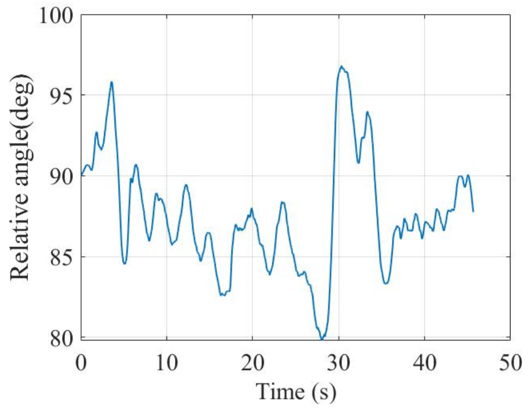
Table 5.5: MAE of relative positions after turning.

Vehicle# / Orientation[°]	1	2	3	4
Yaw	1.0761	0.7887	0.9335	0.8803

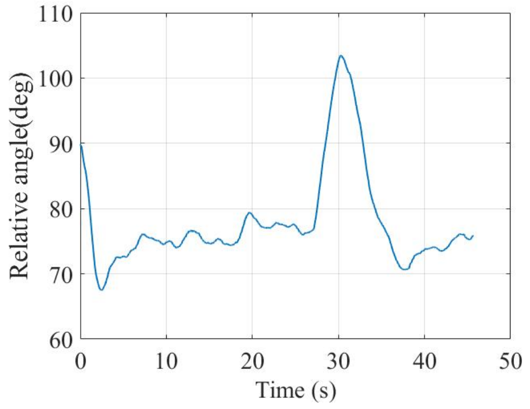
Table 5.6: MAE of orientations after turning.



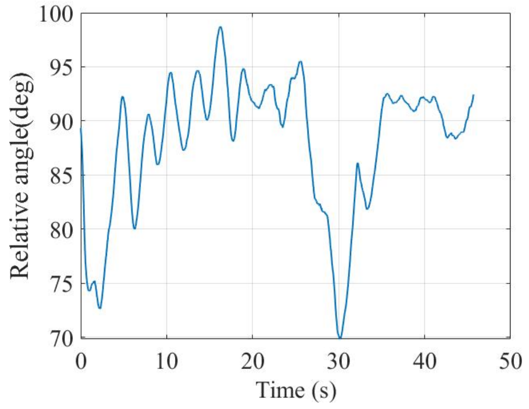
(a)



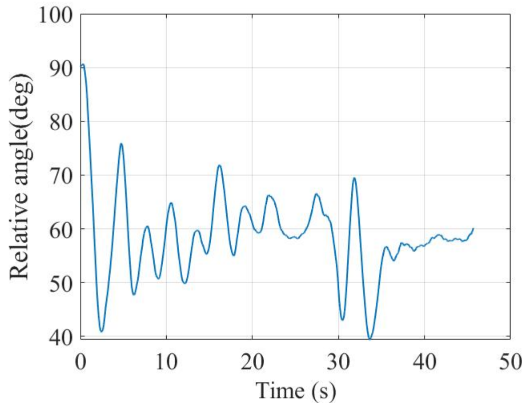
(b)



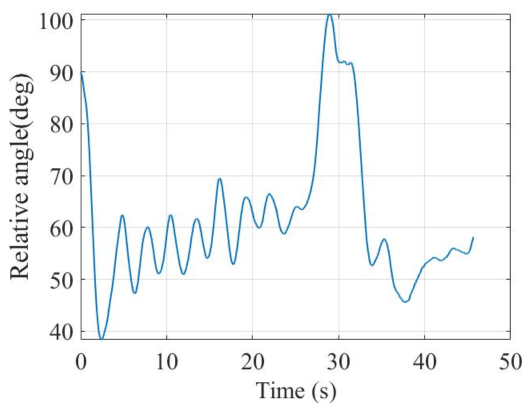
(c)



(d)



(e)



(f)

Figure 5.11: Relative angle between different vehicles (a) vehicle1&2 (b) vehicle1&3 (c) vehicle1&4 (d)vehicle3&1 (e)vehicle3&2 (f) vehicle3&4

A final pattern of the relative position is shown in the Fig5.12, which is coincident with the objective pattern.

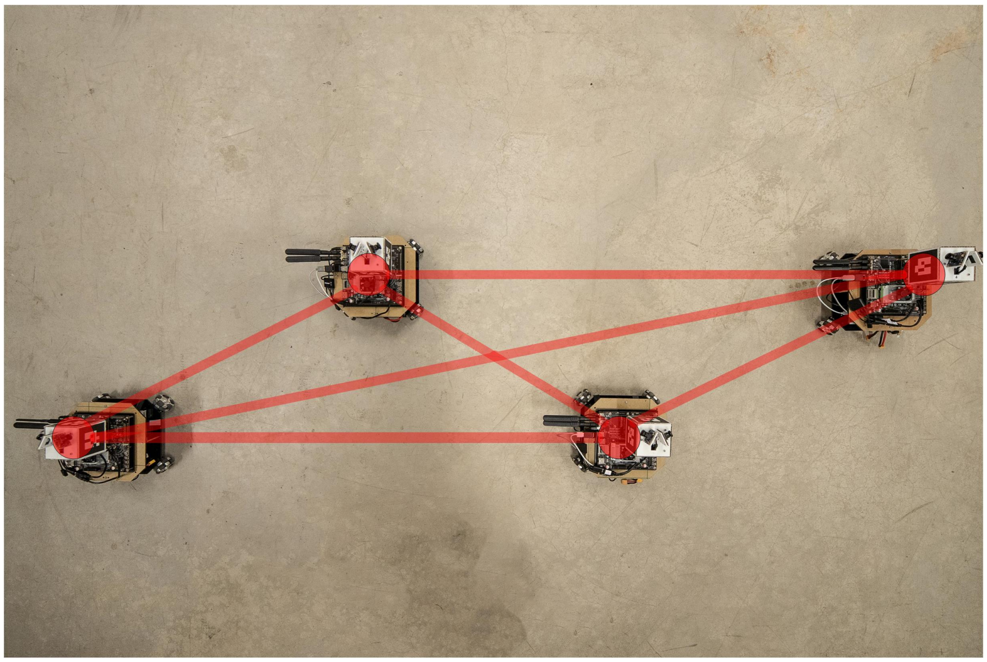


Figure 5.12: Final Formation Pattern

5.4 Conclusion

From these two experiments, it can be seen the formation can converge to and maintain a desired pattern with the connected initial graph. In section 5.3 The velocities and orientation angles of all vehicles can also synchronize to the objective value v_0 and θ_f. During this process, the visual constraint also can be satisfied and no collisions occur among the vehicles. Overall this distributed system can solve the restriction formation problem.

From the experiment results in graph Fig.5.8 to Fig.5.11, we can notice there are some fluctuations and steady errors during the formation process. The reasons might come from several places. The fluctuation may from the residual of gradient at the global minimum of potential function, and insufficient control frequency. The stabilizing error may mainly from the visual system error, recording error of odometer, and accumulative error of IMU.

Chapter 6

Conclusion

6.1 Overall

A gradient-based control method for formation problem has been developed which converges well, provides synchronized orientations and velocities, while keeps visual connections. This provides a base upon which can develop some high-level applications like surveillance, multi-agent cooperation.

Currently, the algorithm shows that it is feasible for a distributed multi-agent system to fulfill self-driving formation under visual restriction. Moreover, a part of nature in this thesis is to give a generalized framework to solve a series of these problems, which is an optimization task under the special restriction or some certain constraints. A basic conceptualization will be introduced in the Section6.2.5.

The major implementation drawback is the steady error and fluctuation. Improving this situation can be a future work on using more powerful hardware and improving algorithm. Such as a more accuracy camera calibration method can decrease visual error, a more powerful on-board computer can reduce the imaging processing period.

6.2 Future Works

6.2.1 Precise Control Algorithm and Accuracy

Now the distributed control algorithm implemented on each vehicle is absolutely independent. A synchronized control strategy may be worth to have a try on low control frequency and computation power limitation situation. The computing can be still finished on the vehicles individually, meanwhile, some synchronized method can be added to unify the motion of whole group. The most intuitional way is adding the synchronized clocks[34].

6.2.2 Relaxing the Connection Constraint

Currently, the connection between vehicles is a strong fully connected structure. It is not necessary to make and maintain the formation with such a strong condition. The pattern will has more variety if the assumption of fully connection can be relaxed. Thus, some vehicles can stay in the blind zone of others. Such a relaxation should be possible to use the existing gradient based controller essentially unmodified, requiring only some modification on the definition of sub graph $\mathscr{G}_{sub}$ and the graph $\mathscr{G}$ which are used for calculating C_{ij} and V_{ij} .

6.2.3 Non-artificial feature based Localization

It is desirable for the formation system to recognize and localization based on the nature feature or object rather than the artificial markers. Such an adaption is feasible with the neural network and deep learning technique[35]. It will definitely be high computation consuming especially using high resolution cameras.

6.2.4 Aerospace Vehicle Formation

Aerial vehicles can provide more flexibility for variety of tasks and have the 3D motion capability. It will be a challenge to develop a neat distributed formation strategy in 3D scenario. Lie group and Lie[36] algebra might be a good way to characterize this kind of problems.

6.2.5 Basis of Formation Problem

The ultra goal should be developing a framework which allows solving most of the formation problem under some certain constraint scenario with optimization methods.

Appendix A

Equipment

A.1 on-board computer –Nvidia TX2

The on-board computer is responsible for those computation consuming calculations, which may include processing sensor information, proposed algorithm calculation and network connection. Here the Nvidia TX2 are used in the project which is a power-efficient high performance embedded device with GPU inside.

Figure A.1: The Nvidia TX2

A.2 Low-level Microcontroller

The low-level controller used in the project is a STM32 based customized controller, which embedded the TB6612FNG motor drive chip, power module and USB series convert chip. The microcontroller board response for resolving the control command and converting to the speed on each motor,meanwhile, calculate the real velocity based on the encoder signal.

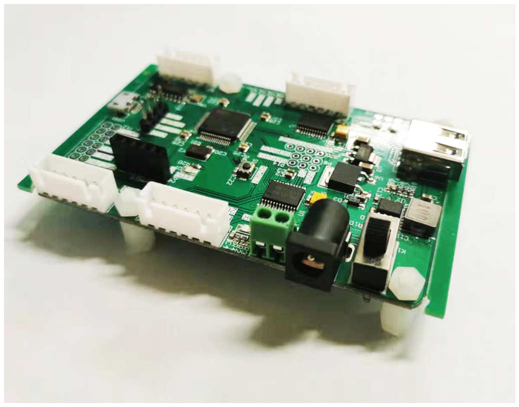


Figure A.2: The Micro-controller

A.3 Camera

Cameras are used to capture image which contain the information of environments. The camera module used in the project is ELP full HD camera with 170° ultra-wide lens. The camera support USB connection with 60 fps in 1280x720 resolution,30 fps in 1920x1080.

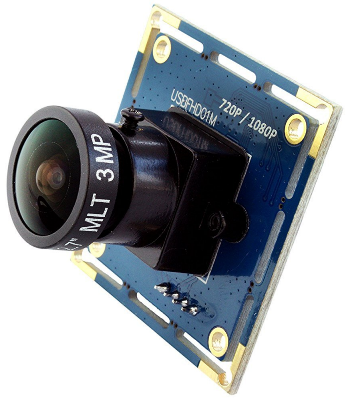


Figure A.3: The Camera Module

A.4 IMU

Inertia Measurement Unit(IMU) include Compass/Gyroscope/Accelerometer which is used to estimate the pose of vehicles. Two kind of IMU sensors are used in the project: VMU931 and phidgets Spatial.

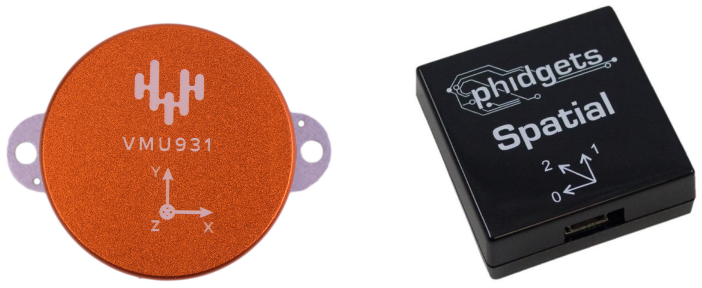


Figure A.4: The Inertia Measurement Unit

Appendix B

Code

All source codes are managed at https://github.com/Alvintang6/robot_formation.

For any question, you are welcomed to email tang12y@uwindsor.ca.

Schematic

The schematic for our MCU is shown in Fig.B.1.

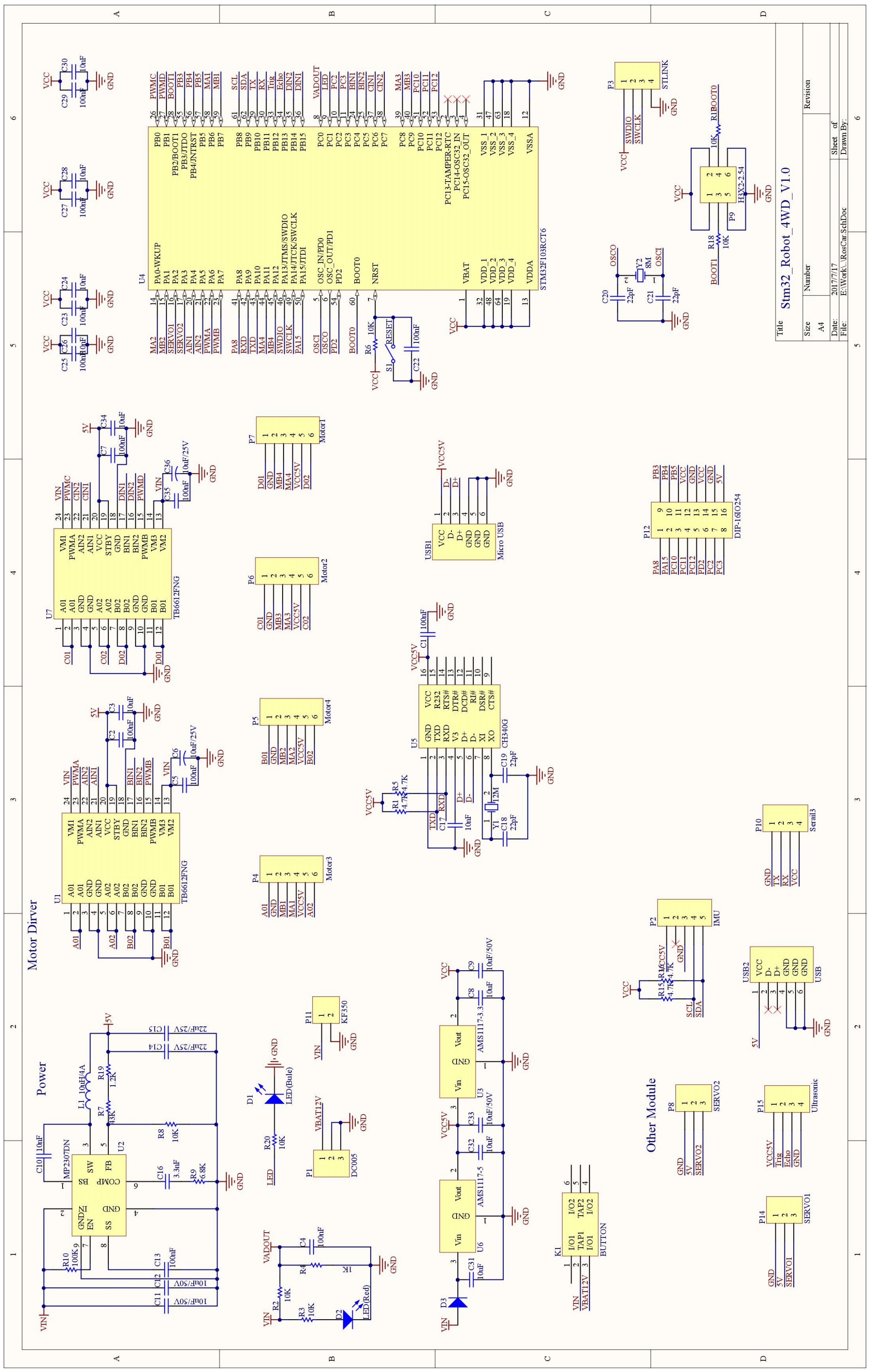


Figure B.1: Schematic of MCU

Appendix C

Stability Proof

The stability proof is imported from[23] and[24] by using Lyapunov stability theory. Taking the following Lyapunov candidate function

$$W = J + 8\sum_{i=1}^{N} \sin^2\left(\frac{\theta_i - \theta_f}{4}\right). \tag{C.1}$$

Define a moving virtual reference frame with the following dynamics:

$$\begin{cases} \dot{x}_0 = v_0 \cos(\theta_f) \\ \dot{y}_0 = v_0 \sin(\theta_f) \\ \dot{\theta}_f = 0 \end{cases} \tag{C.2}$$

The position of i_{th} vehicle regarding with the virtual reference frame C.2 is

$$\tilde{p}_i = \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_f & \sin\theta_f \\ -\sin\theta_f & \cos\theta_f \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} = S(\theta_f)\left(\begin{bmatrix} x_i - x_0 \\ y_i - y_0 \end{bmatrix}\right) \tag{C.3}$$

Taking the derivative of $\tilde{p}_i$ have

$$\begin{aligned}\dot{\tilde{p}}_i = \begin{bmatrix}\dot{\tilde{x}}_i \\ \dot{\tilde{y}}_i\end{bmatrix} &= S(\theta_f)\left(\begin{bmatrix}\cos\theta_i \\ \sin\theta_i\end{bmatrix} v_i - \begin{bmatrix}\cos\theta_f \\ \sin\theta_f\end{bmatrix} v_0\right) \\ &= S(\theta_f)\left(\begin{bmatrix}\cos\theta_i \\ \sin\theta_i\end{bmatrix} \tilde{v}_i + \begin{bmatrix}\cos\theta_i - \cos\theta_f \\ \sin\theta_i - \sin\theta_f\end{bmatrix} v_0\right)\end{aligned} \tag{C.4}$$

Here the $\tilde{v}_i = v_i - v_0$.

At first consider the case,$\mathscr{E}_{sub}(0) = \mathscr{E}_f$, which means the initial graph has all our desired connection inside.

$$\begin{aligned}\dot{J} &= \sum_{i=1}^{N}(\{\nabla_{\tilde{p}_i} J\} \cdot K[\dot{\tilde{p}}_i]) \\ &\subset \sum_{i=1}^{N}\left(\left[\frac{\partial J}{\partial x_i}\frac{\partial x_i}{\partial \tilde{x}_i} + \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial \tilde{x}_i} \;\; \frac{\partial J}{\partial x_i}\frac{\partial x_i}{\partial \tilde{y}_i} + \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial \tilde{y}_i}\right] \cdot K[\dot{\tilde{p}}_i]\right) \\ &\subset \sum_{i=1}^{N}\left(\left(\frac{\partial J}{\partial x_i}\cos\theta_i + \frac{\partial J}{\partial y_i}\sin\theta_i\right) K[\tilde{v}_i]\right. \\ &\quad \left. + \left(\frac{\partial J}{\partial x_i}(\cos\theta_i - \cos\theta_f) + \frac{\partial J}{\partial y_i}(\sin\theta_i - \sin\theta_f)\right) v_0\right)\end{aligned} \tag{C.5}$$

where $K[f](s)$ is Filipov set-valued mapping and satisfies $K[sgn(s)](s) = |s|$.

$$\begin{aligned}\dot{J} \subset \sum_{i=1}^{N}&\left(-\left|\frac{\partial J}{\partial x_i}\cos(\theta_i) + \frac{\partial J}{\partial y_i}\sin(\theta_i)\right| \|\nabla_i J\|\right. \\ &\left. + \left(\frac{\partial J}{\partial x_i}(\cos\theta_i - \cos\theta_f) + \frac{\partial J}{\partial y_i}(\sin\theta_i - \sin\theta_f)\right) v_0\right)\end{aligned} \tag{C.6}$$

$$
\begin{aligned}
\dot{W} &= \dot{J} + \frac{\partial\left(4 - 4\sum_{i=1}^{N}\cos(\frac{\theta_i-\theta_f}{4})\right)}{\partial\theta_i}\dot{\theta}_i \\
&= \dot{J} + 2\sum_{i=1}^{N}\sin\frac{\theta_i-\theta_f}{2}\dot{\theta}_i \\
&\subset \sum_{i=1}^{N}\left(-\left|\frac{\partial J}{\partial x_i}\cos(\theta_i) + \frac{\partial J}{\partial y_i}\sin(\theta_i)\right|\|\nabla_i J\|\right. \\
&\quad \left. -2\sin^2\left(\frac{\theta_i-\theta_f}{2}\right)\right) \leq 0
\end{aligned} \tag{C.7}
$$

Where $\dot{\theta}_i = \omega_i$ is shown in the equation(3.7). Obviously, $W \geq 0$ and its derivative $\dot{W}$ is differentiable and non-positive, which prove the convergence of each $(i,j) \in \mathscr{E}_{sub}$. If $\mathscr{E}_0 \subset \mathscr{E}^f$ it can be reduced to the case that switches of $\mathscr{E}_{sub}$ add new edges.

Bibliography

[1] M. A. Dehghani and M. B. Menhaj, “Communication free leader–follower formation control of unmanned aircraft systems,” *Robotics and Autonomous Systems*, vol. 80, pp. 69–75, 2016.

[2] M. A. Lewis and K.-H. Tan, “High precision formation control of mobile robots using virtual structures,” *Autonomous Robots*, vol. 4, pp. 387–403, Oct 1997.

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

[4] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, “Leader–follower formation control of nonholonomic mobile robots with input constraints,” *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.

[5] J. P. Desai, J. P. Ostrowski, and V. Kumar, “Modeling and control of formations of nonholonomic mobile robots,” 2001.

[6] N. H. M. Li and H. H. T. Liu, “Formation uav flight control using virtual structure and motion synchronization,” in *2008 American Control Conference*, pp. 1782–1787, June 2008.

[7] W. Ren and R. Beard, “Virtual structure based spacecraft formation control with

formation feedback," in *AIAA Guidance, Navigation, and control conference and exhibit*, p. 4963, 2002.

[8] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926–939, Dec 1998.

[9] D. Xu, X. Zhang, Z. Zhu, C. Chen, and P. Yang, "Behavior-based formation control of swarm robots," *mathematical Problems in Engineering*, vol. 2014, 2014.

[10] V. Ayala, J.-B. Hayet, F. Lerasle, and M. Devy, "Visual localization of a mobile robot in indoor environments using planar landmarks," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 1, pp. 275–280, IEEE, 2000.

[11] M. B. Darling, "Autonomous close formation flight of small uavs using vision-based localization," 2014.

[12] M. Saska, J. Vakula, and L. Přeućil, "Swarms of micro aerial vehicles stabilized under a visual relative localization," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3570–3575, IEEE, 2014.

[13] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the pnp problem: A fast, general and optimal solution," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2344–2351, 2013.

[14] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation," in *European Conference on Computer Vision*, pp. 205–220, Springer, 2016.

[15] S. Guler, "Adaptive formation control of cooperative multi-vehicle systems," 2015.

[16] R. Balakrishnan and K. Ranganathan, *A textbook of graph theory.* Springer Science & Business Media, 2012.

[17] D. B. West *et al.*, *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River, NJ, 1996.

[18] H. J. G. de Marina Peinado, *Distributed formation control for autonomous robots.* University of Groningen Groningen, 2016.

[19] K.-K. Oh and H.-S. Ahn, "Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 12, pp. 1809–1820, 2014.

[20] P.-A. Absil and K. Kurdyka, "On the stable equilibrium points of gradient systems," *Systems & control letters*, vol. 55, no. 7, pp. 573–577, 2006.

[21] "What is camera calibration." https://nl.mathworks.com/help/vision/ug/camera-calibration.html. 1994-2019 The MathWorks, Inc.

[22] P. Sturm, *Pinhole Camera Model*, pp. 610–613. Boston, MA: Springer US, 2014.

[23] X. Li, Y. Tan, I. Mareels, and X. Chen, "Compatible formation set for uavs with visual sensing constraint," in *2018 Annual American Control Conference (ACC)*, pp. 2497–2502, June 2018.

[24] X. Li, Y. Tan, J. Fu, and I. Mareels, "On v-shaped flight formation of bird flocks with visual communication constraints," in *2017 13th IEEE International Conference on Control Automation (ICCA)*, pp. 513–518, July 2017.

[25] "About ros." https://www.ros.org/about-ros/. 2018 - Open Source Robotics Foundation, Inc.

[26] J.-B. Song and K.-S. Byun, "Design and control of an omnidirectional mobile robot with steerable omnidirectional wheels," in *Mobile Robotics, Moving Intelligence*, IntechOpen, 2006.

[27] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: a practical introduction to the Robot Operating System.* " O'Reilly Media, Inc.", 2015.

[28] "Augmented reality/3d tracking alvar library." http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/index.html. VTT Technical Research Centre of Finland Ltd.

[29] "Camera calibration." http://wiki.ros.org/image_pipeline/CameraInfo. 2018 - Open Source Robotics Foundation, Inc.

[30] "Camera calibration." http://wiki.ros.org/gscam. 2018 - Open Source Robotics Foundation, Inc.

[31] "Rosbag." http://wiki.ros.org/rosbag. 2018 - Open Source Robotics Foundation, Inc.

[32] "Imu complementary filter." http://wiki.ros.org/imu_complementary_filter?distro=melodic. 2018 - Open Source Robotics Foundation, Inc.

[33] "Ros serial port communication." http://wiki.ros.org/rosserial. 2018 - Open Source Robotics Foundation, Inc.

[34] B. Liskov, "Practical uses of synchronized clocks in distributed systems," *Distributed Computing*, vol. 6, no. 4, pp. 211–219, 1993.

[35] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[36] B. Hall, *Lie groups, Lie algebras, and representations: an elementary introduction*, vol. 222. Springer, 2015.

Vita Auctoris

Name:	Jie Tang
Place of Birth:	Beijing, China
Year of Birth:	1994
Education:	High School (Beijing National Day School,Beijing) 2009-2012
	Bachelor's Degree (North China Electrical Power University, Beijing) 2012-2016
	Master's Degree (University of Windsor, Windsor) 2016-2019