LONG-DISTANCE AND BROAD-BAND AERIAL COMMUNICATION USING DIRECTIONAL ANTENNAS: THEORY, IMPLEMENTATION, AND APPLICATIONS

by

SONGWEI LI

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

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2019

LONG-DISTANCE AND BROAD-BAND AERIAL COMMUNICATION USING DIRECTIONAL ANTENNAS: THEORY, IMPLEMENTATION, AND APPLICATIONS

The members of the Committee approve the doctoral
dissertation of Songwei Li

Yan Wan
Supervising Professor ______________________________

Frank L. Lewis ______________________________

Qilian Liang ______________________________

Sungyong Jung ______________________________

loannis D. Schizas ______________________________

Dean of the Graduate School ______________________________

To My Wife Zhan Xu and My Daughter Skylar Li

ACKNOWLEDGEMENTS

I would like to offer my sincerest gratitude to my supervising professor Prof. Yan Wan, who has supported me throughout my Ph.D. study with her great patience, encouragement and guidance. My sincere thanks also go to the rest of my dissertation committee: Prof. Frank L. Lewis, Prof. Qilian Liang, Prof. Sungyong Jung and Prof. loannis D. Schizas for their insightful comments, encouragement and time to serve in my dissertation committee.

I would also like to thank all of my friends and labmates: Chenyun He, Yixin Gu and Mushuang Liu for their accompany, support, and valuable discussions.

Finally, I am also extremely grateful to my parents, brother, sister and wife for their sacrifice, encouragement and patience.

This dissertation is supported by National Science Foundation (NSF) under grant numbers 1730675, 1714519, 1522458 and the Office of Naval Research (ONR) Grant N00014-18-1-2221.

July 30, 2019

ABSTRACT

LONG-DISTANCE AND BROAD-BAND AERIAL COMMUNICATION USING DIRECTIONAL ANTENNAS: THEORY, IMPLEMENTATION, AND APPLICATIONS

Songwei Li, Ph.D.

The University of Texas at Arlington, 2019

Supervising Professor: Yan Wan

Unmanned aerial vehicles (UAV) have found broad civilian applications. However, existing commercial usages are limited to single UAVs. To facilitate commercial multi-UAV applications, robust UAV-to-UAV communication with long-distance and broad-band capabilities is critical. Such a communication architecture should not rely on ground infrastructure support, and hence can be applied whenever needed in scenarios where infrastructures are not available. It is foreseen to have extensive usages in a wide spectrum of commercial UAV applications such as emergency response, remote sensing, and nondestructive health monitoring.

A promising solution for such a communication architecture is aerial communication using directional antennas (ACDA). We developed several preliminary ACDA solutions. This dissertation presents improvements to the ACDA system, in terms of theory, implementation, and applications. With respect to theory, a critical component of the ACDA system is the automatic alignment of directional antennas to maximize communication performance. To account for an unstable global positioning system GPS environment, we develop a novel stochastic optimal control solution that integrates reinforcement learning (RL), an effective uncertainty evaluation method called multivariate probabilistic collocation method (MPCM), and an unscented Kalman filter (UKF) for the nonlinear random switching dynamics. Part of this theoretical solution is implemented. With respect to implementation, we redesign the hardware and software to improve robustness, throughput and endurance. Two versions of the ACDA system are implemented. The first version includes a complete new design of platform, communication, computing, control, middleware, and interface components. It features a communication and control co-design to achieve robust performance in a GPS-denied environment. The second version is upgraded with a UAV platform, computing component, battery solution, and rotation structure to reduce interference and improve endurance. Beyond the visual line of sight UAV control using the ACDA system delay is also implemented. With respect to applications, built upon the ACDA system, we first develop a UAV-carried vision-based monitoring system that allows a UAV to continuously track and monitor a mobile infrastructure and transmit back the monitoring information in real-time from a remote location. We then develop a leader-follower tracking system using the ACDA system, that enables cooperative UAV control over a long distance, which can be used in broad multi-UAV remote sensing, monitoring, and emergency response applications.

TABLE OF CONTENTS

Chapter Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

Table Page

CHAPTER 1

Introduction

1.1 Background

UAVs have found broad commercial use in civilian applications [1,2]. UAVs have the capability to reach dangerous zones without human access and their reachability is not limited by terrain constraints. For instance, UAVs find many applications in critical infrastructure monitoring, such as gas emission and oil spills on offshore oil platforms [3] and boilers at power plants [4]. UAV-assisted emergency response has found its significant value in reducing response time and better allocating resources. In [5, 6], UAV-based detection systems were developed for disaster search and rescue. An on-board computing processes images from cameras and detects a target and its location. Paper [7] developed a three-dimensional reconstruction method from UAV images to identify collapsed buildings after earthquakes.

However, existing commercial usages are limited to single UAVs. Multi-UAV systems are more appealing with enhanced capabilities, and are foreseen to have substantial commercial markets in the near future. Paper [8] presents a path planning algorithm to control multiple UAVs to monitor forest fires. In [9], the author implemented a tracking algorithm to control multiple UAVs to map a nuclear radiation field. In [10], a UAV-based Internet of things (IoT) platform was developed with a long-term evolution (LTE) network to conduct crowd surveillance using facial recognition tools. To support an increasing number of commercial multi-UAV applications, robust UAV-to-UAV communication is critical to exchange information, main safe maneuvering, and cooperate to achieve time-critical missions.

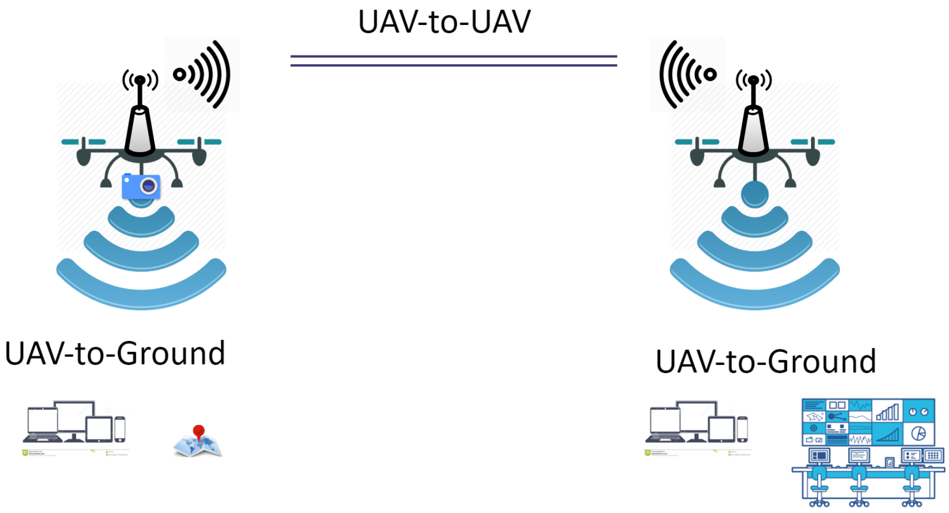


Figure 1.1. Illustration of the broadband long-distance communication infrastructure using controllable UAV-carried directional antennas..

A promising solution for such a communication architecture is aerial communication using directional antennas (ACDA) (see Figure 1.1). Compared to the omnidirectional communication solutions, the use of directional antennas allows the energy to focus along a certain direction, and thus offers various benefits such as extended communication distance, reduced antenna power consumption, and reduced communication interference. We developed several preliminary ACDA solutions. Paper [11] verifies the feasibility of long-distance and broadband UAV communication through directional antennas using experimental studies, and developed an early version of the ACDA system in which UAV locations are fixed. To enable a robust communication channel under UAV mobility, the autonomous alignment of antenna directions is needed. Therefore, in [12], a proportional-integral-derivative (PID) controller was designed to align antenna headings under wind disturbances based on the GPS signals transmitted through the omni control and command channel. In [13], a complete ACDA system was designed and implemented, with detailed descriptions of the UAV platform configuration, onboard autopilot, communication system, and antenna heading control. For the antenna heading control, a linear-quadratic-gaussian (LQG) controller was developed to minimize a quadratic GPS-based tracking error. An initial scan algorithm that achieves automatic antenna direction alignment without the presence of GPS signals was also developed. In [14], the received signal strength indication (RSSI) was used as an additional measurement signal to compensate GPS signals for the automatic alignment of directional antennas, and a proportional-integral (PI) controller is used for antenna control.

1.2 Overview of this Dissertation

This dissertation presents a suite of improvements to the ACDA system. We here summarize the contributions of this dissertation, in terms of theory, implementation, and applications.

With respect to theory, a novel stochastic optimal control solution is developed that integrates reinforcement learning, a multivariate probabilistic collocation method, and an unscented Kalman filter (UKF) for the nonlinear random switching dynamics in unstable GPS and unknown communication environments. Part of this theoretical solution is implemented on the automatic alignment of directional antennas to maximize communication performance.

With respect to implementation, we redesign the hardware and software to improve robustness, throughput and endurance. Two new versions of the ACDA system are implemented. The first version includes a complete new design of platform, com munication, computing, control, middleware, and interface components. It features a communication and control co-design to achieve robust performance in a GPS-denied environment. The second version is upgraded with a UAV platform, computing component, battery solution, and rotation structure to reduce interference and improve endurance. Beyond visual line of sight UAV control using the ACDA system relay is also implemented.

With respect to applications, built upon the ACDA system, we first develop a UAV-carried vision-based monitoring system that allows a UAV to continuously track and monitor a mobile infrastructure and transmit back the monitoring information in real-time from a remote location. We then develop a leader-follower tracking system using the ACDA system, that enables cooperative UAV control over long distance, which can be used in broad multi-UAV remote sensing, monitoring, and emergency response applications. My Ph.D. study has resulted in publications [13–22].

1.3 Organization of This Dissertation

Chapter 2 discusses the development of reinforcement learning (RL)-based online directional antenna control solution for the ACDA system in uncertain and varying UAV movement patterns and unstable GPS and unknown communication environments. The solution adopts a random switching UAV mobility model to capture the uncertain intentions of UAVs executing surveillance-like missions for better tracking. To account for an unstable GPS environment, the control solution features a learning of communication channel models to provide additional measurement signals in GPS-denied settings. A novel stochastic optimal control solution is developed that integrates RL, an effective uncertainty evaluation method called multivariate probabilistic collocation method (MPCM), and UKF for the nonlinear random switching dynamics. Simulation studies are conducted to illustrate and validate the proposed solutions. The results are documented in paper [17] accepted by the 2019 IEEE Vehicular Technology Conference.

Chapter 3 describes the design and implementation of a new ACDA system, including the platform, communication, computing, control, middleware, and interface components. The communication component features a directional-antenna equipped UAV-to-UAV channel that shares the transmissions of the application data, control and commands, and the autonomous directional antenna control that focuses the energy and significantly extends the communication distance with reduced interference. In addition, the water-resistant design supports the ACDA system to work in the severe weather. The performance of the ACDA system is verified using simulation studies, field tests, and disaster drills. The results can be found in paper [21] published in IET Control Theory & Applications.

Chapter 4 describes development of a beyond visual line of sight (BVLOS) control solution for a remote UAV, using a local UAV relay equipped with automatically aligned directional antennas. Many new UAV applications require BVLOS control of UAVs. Examples include UAV-based nondestructive health monitoring of civil infrastructures and remote UAV-assisted emergency response. The use of a UAV relay permits the transmission of flight control signals even if blockages exist between the UAV and the operator. The use of directional antennas extends the UAV control distance to kilometers with high data throughput. Built upon our previously developed ACDA system, we first upgrade the hardware and software to improve the throughput and endurance. We then design and implement a BVLOS control solution that includes a two-way relay communication. In one direction, low-resolution video views from the remote UAV are transmitted through the local UAV relay to help the UAV operator control the remote UAV. In the other direction, UAV control signals are relayed through the local UAV and transmitted to the remote UAV for BVLOS navigation. The results are documented in paper [19] submitted to the 2019 IEEE GLOBECOM Workshop on Computing-Centric Drone Networks.

Chapter 5 describes the development of a UAV-carried vision-based monitoring system that allows a UAV to continuously track and monitor a mobile infrastructure and transmit the monitoring information back in real-time from a remote location. The monitoring system uses a simple UAV-mounted camera and requires only a single feature located on the mobile infrastructure for target detection and tracking. The computation-effective vision-based tracking solution based on a single feature is an improvement over existing vision-based lead-follower tracking systems that either have poor tracking performance due to the use of a single feature, or have improved tracking performance at a cost of the usage of multiple features. In addition, a UAV-carried aerial networking infrastructure using directional antennas is used to enable robust real-time transmission of monitoring video streams over a long distance. Automatic heading control is used to self-align headings of directional antennas to enable robust communication in mobility. Compared to existing omni-communication systems, the directional communication solution significantly increases the operation range of remote monitoring systems. The result can be found in paper [15] published in 2017 SPIE on Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure.

Chapter 6 discusses the development of a leader-follower tracking control system based on the ACDA system, because cooperative multiple UAVs can improve coverage with effiency in applications such as remote sensing and precision agriculture. The ACDA system is first updated with the follower and leader UAVs connected to the ACDA system using a robot operation system (ROS). Flight control messages, sensor data, and flight status data are transmitted in the ACDA system. Then a feedback control is developed to drive the follower UAV track the leader UAV. Simulation studies verify the leader-follower tracking system.

CHAPTER 2

Learning and Uncertainty-Exploited Directional Antenna Control for Robust Aerial Networking

2.1 Introduction

Aerial networking using directional antennas is a promising solution to enable long-distance and broad-band UAV-to-UAV communication [11–13, 23]. Through using directional antennas that focus the transmission energy in certain direction, ACDA significantly extends the communication distance and rejects interference, compared to omni-drectional antenna based solutions. With ACDA, UAVs carried communication infrastructures can be quickly deployed to deliver Wi-Fi services from the air, through which high-rate data such as monitoring streams from remote locations can be transmitted in real-time (see Figure 1.1). The application of such a system spans multiple fields, such as remote large-area surveillance, remote infrastructure health monitoring, and the provision of on-demand emergency communication services [13, 15, 23].

A critical component of the ACDA system is the automatic alignment of directional antennas to maximize the communication performance. Each UAV in the ACDA system carries a rotational plate mounted with a directional antenna [13], which is controlled to align with the directional antenna carried by the other UAV. Robust automatic alignment of directional antennas is not easy to achieve, considering practical issues such as the limited sensing devices due to the physical constraints of UAV payload and power supplies, uncertain and varying UAV mobility, and unstable GPS and unknown communication environments.

There are two general design configurations of the ACDA system, depending on whether the communication channel used for antennas control is omni or not. The first configuration uses a directional antenna- equipped broad-band channel for the transmission of application-oriented data (e.g., real-time video streams), and an additional low-rate omni-directional communication channel for controls and commands [13]. However, the omni-directional control channel suffers from practical issues such as interference and dissipation over a long communication distance. As such, we aim to design using the second configuration where the high-rate application data and low-rate controls and commands share the same channel equipped with directional antennas.

Although more practical, this solution that removes the additional controls and commands channel introduces more challenges to the robustness requirement of antennas control. As the controls and commands cannot be transmitted if the directional communication channel fails, we require the antennas control system to be able to lock and track the other directional antennas, once the communication channel is established initially. To do that, we develop a mobility framework that uses random mobility models (RMMs) [24–26] to capture the uncertain intentions of UAV maneuvers. RMMs describe the uncertain movement characteristics of mobile agents, and play critical roles in networking studies. Please see the the survey [25] for RMMs that are suitable for various UAV applications. RMMs are random switching models that require new development for optimal control.

In addition, in indoor and many emergency scenarios, GPS signals may be unstable considering the environmental disturbances and blockages. In GPS unstable or denied environment, we need additional measurement signals for antenna control. In [14], we adopted Received Signal Strength Indicator (RSSI), a communication performance indicator, to compensate unstable GPS signals, under the assumption that the communication environment is perfect. However, in real applications, the Wi-Fi communication channel model is strongly affected by its environment and hence less likely to be perfect, considering the reflection, refraction and absorption by buildings, obstacles, and interference sources. In this chapter, we develop a control solution that learns directional Wi-Fi channel models online and provides RSSI as an alternative measurement signal in a GPS-denied setting.

Our antenna control adopts a novel stochastic optimal control approach that integrates Reinforcement Learning for online optimal control, Multivariate Probabilistic Collocation Method for effective uncertainty evaluation, and Unscented Kalman Filter for nonlinear state estimation. On the aspect of optimal control, studies on the RL in optimal control problems have been developed in [27,28] with deterministic system dynamics. Papers [29, 30] developed the stochastic optimal control solution that integrates MPCM and RL methods for systems modulated by uncertainties. In these papers, the uncertainties are relatively simple, as compared to the more complicated random switching RMMs considered in this chapter. On the aspect of estimation, nonlinear system estimation methods such as Extended Kalman Filter (EKF) and UKF have been widely used typically for known and deterministic systems corrupted with additive noises, instead of the random switching RMMs. Due to these limitations, in this chapter we develop a new stochastic optimal control solution for systems that involve nonlinear random switching RMMs and limited measurements, by integrating UKF and RL with MPCM.

This chapter is organized as follows. In Chapter 2.2, we describe the ACDA system shown in Figure 1.1, including both system models and measurement models. The antenna control problem is also formulated. In Chapter 2.3, we propose the RL based stochastic optimal control solutions when 1) the RSSI model is unknown but the GPS signal is available, and 2) the RSSI model is learned in either GPS-available or GPS-denied environments. In Chapter 2.4, simulation studies are conducted.

2.2 Modeling and Problem Simulation

In this section, we first describe the ACDA system model, including the UAV RMM and directional antenna dynamics. We then describe the two measurement models using GPS and RSSI. The antenna control problems are then formulated.

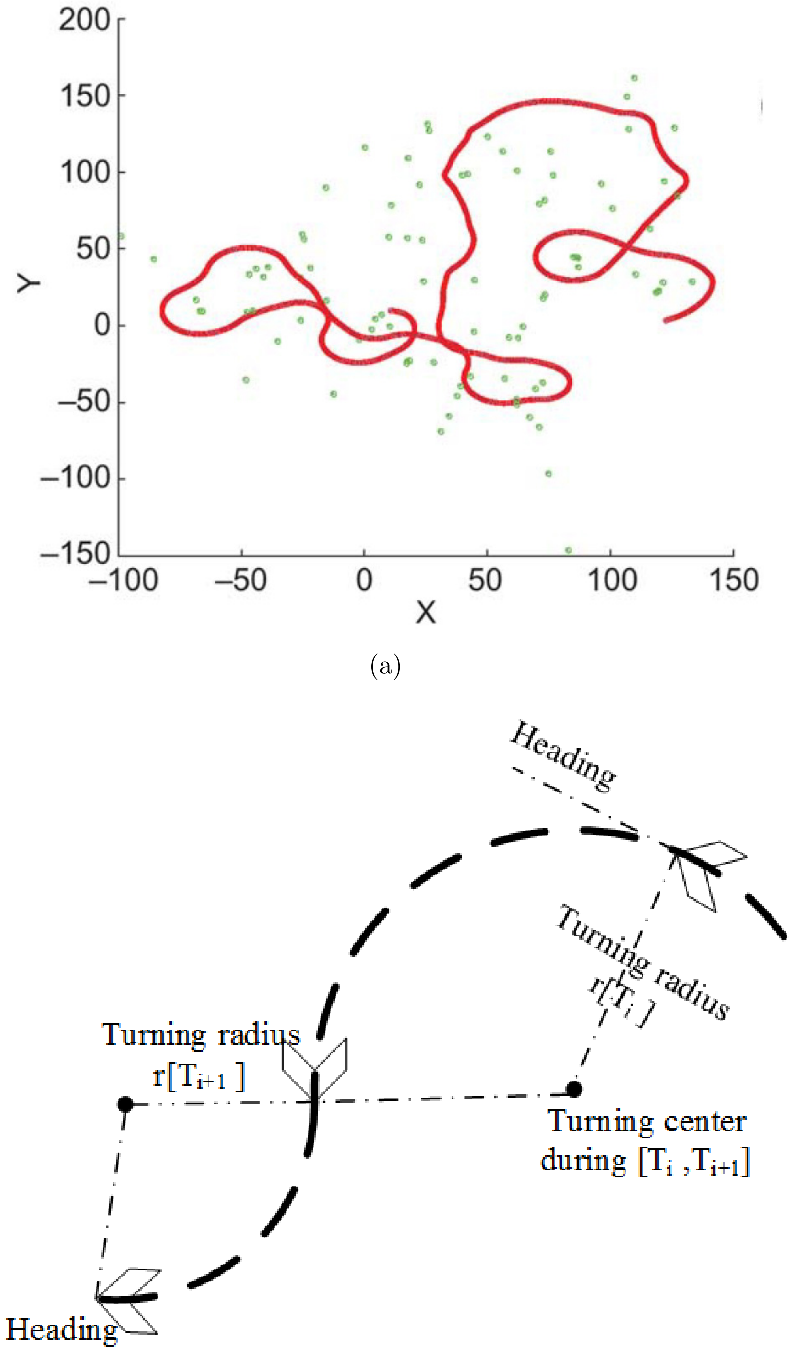
2.2.1 System Models

We consider two UAVs independently fly in a low-altitude airspace at approximately the same height to fulfill their missions such as search and rescue (see Figure 1.1). On each UAV, a tunable plate attached with a directional antenna is installed and driven by a gear motor [13]. To establish a long-range air-to-air communication channel to transmit both application data (e.g., surveillance videos) and controls and commands, the communication performance of this channel needs to be maximized.

2.2.1.1 UAV Random Mobility Model

We here use the smooth turn (ST) mobility model ([24, 26]) to capture the uncertain intentions of UAVs executing surveillance-like missions (see Figure 2.1(b)).

The random maneuvers described by a ST mobility model work as follows. At randomly selected points T_0^i, T_1^i, $T_2^i, \cdots$, where $0 = T_0^i < T_1^i < \cdots$, UAV i selects a point in the airspace along the line perpendicular to its current heading direction, and then circles around it until the UAV chooses another turning center. The perpendicularity guarantees the smoothess of trajectories [24]. The time duration



(a)

(b)

Figure 2.1. Illustration of the ST RMM: (a) UAV trajectory ensemble (red curve). Green spots are the randomly chosen turning centers; (b) maneuver selection and switching.

for UAV i to maintain its current maneuver $\tau_i[T_j^i] = T_{j+1}^i - T_j^i$ follows an exponential distribution as motivated by its memoryless property [31].

$$f_\tau(\tau_i[T_j^i]) = \lambda_i e^{-\lambda_i \tau_i[T_j^i]}, \tag{2.1}$$

where $1/\lambda_i$ is the mean of $\tau_i[T_j^i]$. The velocity $v_i[T_j^i]$ follows a uniform distribution, and the inverse of the turning radius $\frac{1}{r_i[T_j^i]}$ follows the zero-mean Gaussian distribution with variance σ_i^2.

Denote the positions of the UAV i along x and y axes at time instant k as $x_i[k]$ and $y_i[k]$ respectively. The dynamics of UAV i (denote as $f_i(.)$) following the ST uncertain maneuvering intentions is described as

$$\begin{aligned} x_i[k+1] &= x_i[k] + v_i[k]\cos(\phi_i[k])\delta, \\ y_i[k+1] &= y_i[k] + v_i[k]\sin(\phi_i[k])\delta, \\ \phi_i[k+1] &= \phi_i[k] + \omega_i[k]\delta, \end{aligned} \tag{2.2}$$

where δ is the sampling period, $\phi_i[k]$ and $\omega_i[k]$ are the heading angle and angular velocity at the time instant k, and $\omega_i[k] = \frac{v_i[k]}{r_i[k]}$.

Note that the ST RMM is a random switching model composed of two types of random variables [32]. Type 1 random variables includes $v_i[k]$, and $r_i[k]$. They describe the characteristics for each maneuver, and show a random switching behavior.

$$v_i[k] = \begin{cases} v_i[T_j^i], & \text{if } \exists j \in [0,1,2,..), k = T_j^i \\ v_i[k-1], & \text{if } \forall j = 0,1,2,.., k \neq T_j^i \end{cases} \tag{2.3}$$

$$r_i[k] = \begin{cases} r_i[T_j^i], & \text{if } \exists j \in [0,1,2,..), k = T_j^i \\ r_i[k-1], & \text{if } \forall j = 0,1,2,.., k \neq T_j^i \end{cases} \tag{2.4}$$

The maneuvers' random switching behavior is governed by the type 2 random variable, $\tau_i[T_j^i]$, which describes how often the switching of type 1 random variables occurs.

The two groups of uncertain maneuvers for the UAVs $(v_1[T_j^1], r_1[T_j^1], \tau_1[T_j^1])$ and $(v_2[T_j^2], r_2[T_j^2], \tau_2[T_j^2])$ are independent, as UAV mobility is application-specific, and is not constrained from the communication mission.

2.2.1.2 Directional Antennas Dynamics

The directional antenna installed on each UAV autonomously adjust its heading angle to establish a robust communication channel between the two UAVs. For UAV i, the heading angle dynamics of its directional antennas is described as

$$\theta_i[k+1] = \theta_i[k] + (\omega_i^*[k] + \omega_i[k])\delta, \tag{2.5}$$

where θ_i is the heading angle of antennas i, and ω_i^* is the angular velocity of antennas i due to its heading control. Note that the change of θ_i is caused by both the control of antennas i, ω_i^*, and the movement of UAV i, ω_i.

2.2.2 Measurement Models

We consider two measurement signals for the ACDA system, GPS and RSSI.

2.2.2.1 GPS Measurement

Denoted the measured GPS signal for UAV i as $Z_{G,i}(k)$, then

$$Z_{G,i}[k] = H_G(k)X_i[k] + \varpi_{G,i}[k], \tag{2.6}$$

where $H_G = [1, 0, 0, 0; 0, 1, 0, 0]$ is the measurement matrix, $X_i[k] = [x_i[k], y_i[k], \phi_i[k], \theta_i[k]]^T$ is the system state of UAV i, and $\varpi_{G,i}$ is the white Gaussian noise with zero mean and covariance $R_{G,i}$. Denote the relation between the GPS signals and system states as $h_{G,i}$, i.e., $Z_{G,i}[k] = h_{G,i}(X_i[k])$.

2.2.2.2 RSSI Measurement

Denote the measured RSSI signal as $Z_R[k]$, then according to Friis free space equation [33], one has

$$\begin{aligned} Z_R[k] =& P_{t|dBm}[k] + 20\log_{10}(\lambda) - 20\log_{10}(4\pi) \\ & - 20log_{10}(d[k]) + G_{l|dBi}[k] + \varpi_R[k], \end{aligned} \tag{2.7}$$

where $P_{t|dBm}[k]$ is the transmitted signal power, λ is the wavelength, and $d[k]$ is the distance between the two UAVs at time k. $G_{l|dBi}[k]$ is the sum of gains at both the transmitting and receiving sides [34]. The Ubiquiti NanoStation loco M5 directional antennas [35] that we use in the ACDA system is modeled based on the filed pattern of the end-fire array antennas [36],

$$\begin{aligned} G_{l|dBi}[k] =& (G^{max}_{t|dBi} - G^{min}_{t|dBi}) \\ & \times \sin\frac{\pi}{2n}\frac{\sin\left(\frac{n}{2}(k_a d_a(\cos(\gamma_t[k] - \theta_t[k])) - 1) - \frac{\pi}{n}\right)}{\sin\left(\frac{1}{2}(k_a d_a(\cos(\gamma_t[k] - \theta_t[k])) - 1) - \frac{\pi}{n}\right)} \\ & + (G^{max}_{r|dBi} - G^{min}_{r|dBi}) \\ & \times \sin\frac{\pi}{2n}\frac{\sin\left(\frac{n}{2}(k_a d_a(\cos(\gamma_r[k] - \theta_r[k])) - 1) - \frac{\pi}{n}\right)}{\sin\left(\frac{1}{2}(k_a d_a(\cos(\gamma_r[k] - \theta_r[k])) - 1) - \frac{\pi}{n}\right)} \\ & + G^{min}_{t|dBi} + G^{min}_{r|dBi}, \end{aligned} \tag{2.8}$$

where $G^{max}_{t|dBi}$, $G^{min}_{t|dBi}$, and $G^{max}_{r|dBi}$, $G^{min}_{r|dBi}$ are the maximum and minimum gains of transmitting and receiving antennas. k_a is the wave number, and $k_a = \frac{2\pi}{\lambda}$. n and d_a are design parameters of the directional antenna. $\theta_t[k]$ and $\theta_r[k]$ are the heading angles of the transmitting and receiving antennas at time k, respectively. $\gamma_t[k]$ and $\gamma_r[k]$ are the heading angles of the transmitting and receiving antennas corresponding to the maximal G_l at time k, respectively.

The parameters $G^{max}_{t|dBi}$, $G^{min}_{t|dBi}$,$G^{max}_{r|dBi}$, and $G^{min}_{r|dBi}$, can be obtained from the antenna's datasheet. In ACDA, the two directional antennas are of the same type, and hence $G^{max}_{t|dBi} = G^{max}_{r|dBi}$, and $G^{min}_{t|dBi} = G^{min}_{r|dBi}$. In an imperfect environment (e.g., where disturbances and interference exist), these parameters in $G_{l|dBi}[k]$ can be environment-specific.

Similarly, in a perfect communication environment, $\gamma_t[k]$ and $\gamma_r[k]$ are achieved when the two antennas are aligned [14]. Affected by the impact of imperfect environment, such as blockages, the desired heading angles can be captured as

$$\gamma_r[k] = \arctan\frac{y_t[k] - y_r[k]}{x_t[k] - x_r[k]} + \theta_{r_{env}} \tag{2.9}$$

$$\gamma_t[k] = \arctan\frac{y_r[k] - y_t[k]}{x_r[k] - x_t[k]} + \theta_{t_{env}} \tag{2.10}$$

where $(x_t[k], y_t[k])$ and $(x_r[k], y_r[k])$ are the positions of UAVs that carry the transmitting and receiving antennas respectively, and $\theta_{r_{env}}$ and $\theta_{t_{env}}$ are environment-specific shift angles at the receiver and transmitter sides. $\theta_{r_{env}}$ and $\theta_{t_{env}}$ are zeros in a perfect environment.

2.2.3 Problem Formulation

We aim to design antennas' angular velocities to maximize the expected RSSI performance of ACDA over a look-ahead window. The RSSI model contains unknown environment-specific parameters ($G^{max}_{t|dB}$, $G^{min}_{t|dBm}$, and θ_{env}), and the UAV dynamics contain uncertain parameters $((v_1[k], r_1[k], \tau_1[T^1_j], v_2[k], r_2[k], \tau_2[T^2_j]))$.

Here we formulate the problem as a stochastic optimal control problem. Mathematically, considering the random switching system dynamics described in Equations (2.2) and (2.5), the optimal control policy $U[k]$ is sought to maximize the expected value function

$$V(X[k]) = E\{\sum_{l=k}^{k+N} \alpha^{l-k} Z_R[l](X[l], U[k])\}, \tag{2.11}$$

where $X[k]$ is the global state, $X[k] = [X_1^T[k], X_2^T[k]]^T$. $U[k]$ is the control input, $U[k] = [U_1[k], U_2[k]]^T$, $U_i[k] = [\omega_i^*[k]]$. $Z_R[l]$ is the RSSI signal at time l, and $\alpha \in (0, 1]$

is a discount factor. Note that the control is decentralized, in the sense that each antenna finds its own optimal control policy, with the assumption that the other antenna adopts its optimal control policy.

In the rest of this article, we develop the control solution for one of the UAVs, and denote this UAV as the local UAV, or UAV 1, and the other UAV as the remote UAV, or UAV 2. The control solution for the other UAV is designed in the same manner.

2.3 Reinforcement Learning Based Stochastic Optimal Control for ACDA

In this section, we develop new on-line solutions to solve the stochastic optimal control problem for the ACDA system described in Chapter 2.2.3. Chapter 2.3.1 describes the online solution when GPS is available but the RSSI model is unknown. Chapter 2.3.2 further develops on-line solutions in both GPS-available and GPS-denied environments, with the learned environment-specific RSSI model.

2.3.1 Stochastic Optimal Control with Unknown RSSI

The stochastic optimal control solution includes two main steps: 1) state estimation, and 2) adaptive optimal controller design. GPS signal is needed in this solution to learn the environment-specific RSSI model.

2.3.1.1 State Estimation

The states of both local and remote UAVs need to be estimated. For the remote UAV that has random switching dynamics, the RMM-related maneuvers ($v_2[k]$, $r_2[k]$, and $\tau_2[T_j^2]$) are unknown to the local UAV, and hence the remote UAV's states $(x_2[k], y_2[k], \phi_2[k])$ can not be estimated directly using existing filtering types of methods. We design a new estimation algorithm for the nonlinear and random switching dynamics.

For the local UAV, the trajectory-specific maneuvers ($v_1[k]$, $r_1[k]$, and $\tau_1[T_j^i]$) are known locally, and hence, the self-system states $(x_1[k], y_1[k], \phi_1[k])$ can be estimated utilizing UKF as described in [14, Section 3.1]. We do not repeat the process here due to the limited space.

A critical step in the state estimation of a random switching system is to estimate the expected system state under random switching behaviors. This involves uncertainty evaluation that is typically solved by the Monte Carlo method, which is too slow to be used for on-line solutions. Here, we adopt an efficient uncertainty sampling method, called multivariate probabilistic collocation method [37]. MPCM permits using a very limited number of smartly selected samples to estimate the output mean for a system of input-output mapping subject to uncertain input parameters as described in the following lemma.

Lemma 1. *[37, Theorem 2] Consider a system G modulated by m independent uncertain parameters, a_i, where $i \in \{1, ..m\}$,*

$$G(a_1, ..., a_m) = \sum_{j_1=0}^{2n_1-1} \sum_{j_2=0}^{2n_2-1} ... \sum_{j_m=0}^{2n_m-1} \psi_{j_1,...,j_m} \prod_{i=1}^{m} a_i^{j_i}, \tag{2.12}$$

where a_i is an uncertain parameter with the degree up to $2n_i - 1$. n_i is a positive integer for any i. $\psi_{j_1,...,j_m} \in \mathbb{R}$ are the coefficients. Each uncertain parameter a_i follows an independent pdf $f_{a_i}(a_i)$. The MPCM approximates $G(a_1, ...a_m)$ with the following low-order mapping

$$G'(a_1, ..., a_m) = \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} ... \sum_{j_m=0}^{n_m-1} \Omega_{j_1,...,j_m} \prod_{i=1}^{m} a_i^{j_i}, \tag{2.13}$$

with $E[G(a_1, ..., a_m)] = E[G'(a_1, ..., a_m)]$, where $\Omega_{j_1,...,j_m} \in \mathbb{R}$ are coefficients. MPCM reduces the number of simulations from $2^m \prod_{i=1}^{m} n_i$ to $\prod_{i=1}^{m} n_i$.

Denote the switching behavior of the remote UAV at time k as $s[k]$. $s[k]=1$ or 0 represent the current maneuver switches at time k or not. With the two possible switching behaviors, the expected conditional current state can be derived as

$$\begin{aligned}&E(X_2[k]|X_2[k-1])\\&=E(X_2[k]|X_2[k-1],s[k-1]=0)P(s[k-1]=0)\\&\quad+E(X_2[k]|X_2[k-1],s[k-1]=1)P(s[k-1]=1)\end{aligned} \tag{2.14}$$

Given a previous state $X_2[k-1]$, if UAV 2 does not change its maneuver at time $k-1$, i.e., $s[k-1]=0$, then its current state $E(X_2[k]|X_2[k-1],s[k-1]=0)$ can be estimated using the previous maneuver $v_2[k-1]$ and $r_2[k-1]$ from the system dynamics $f(v_2[k-1],r_2[k-1],X_2[k-1])$. If UAV 2 switches its maneuver at time $k-1$, then the current state estimation is much more complex. Define a system mapping subject to uncertain input parameters $v_2[T_j^2]$ and $r_2[T_j^2]$: $G_2(v_2[T_j^2],r_2[T_j^2],X_2[k-1])=f(v_2[T_j^2],r_2[T_j^2],X_2[k-1])$. When $s[k-1]=1$, the expected current state $E(X_2[k]|X_2[k-1],s[k-1]=1)$ can be estimated from the mean output of the system mapping $G_2(v_2[T_j^2],r_2[T_j^2],X_2[k-1])$, i.e., $E(X_2[k]|X_2[k-1],s[k-1]=1)=E[G_2(v_2[T_j^2],r_2[T_j^2],X_2[k-1])]$, using MPCM according to Lemma 1 and paper [37]. Under the assumption that the two uncertain parameters $v_2[T_j^2]$ and $r_2[T_j^2]$ have a degree up to $2n_1-1$ and $2n_2-1$ respectively, then $G_2(v_2[T_j^2],r_2[T_j^2],X_2[k-1])$ has the following form.

$$G_2(v_2[T_j^2],r_2[T_j^2],X_2[k-1])=\sum_{j_1=0}^{2n_1-1}\sum_{j_2=0}^{2n_2-1}\psi_{j_1,j_2}(X_2[k-1])v_2^{j_1}r_2^{j_2} \tag{2.15}$$

Theorem 1. *Given the previous state $X_2[k-1]$ of the remote UAV 2, the expected current state $E(X_2[k]|X_2[k-1])$ is estimated by the local UAV 1 as*

$$\begin{aligned}&E(X_2[k]|X_2[k-1])\\&= P_2 E[G_2'(v_2[T_j^2], r_2[T_j^2], X_2[k-1])]\\&\quad + (1-P_2) f(v_2[k-1], r_2[k-1], X_2[k-1]),\end{aligned} \tag{2.16}$$

where P_2 is the switching probability of the remote UAV's maneuver at each time instant, $P_2 = \lambda_2\delta$. $G_2'(v_2[T_j^2], r_2[T_j^2], X_2[k-1])$ is a reduced-order mapping of $G_2(v_2[T_j^2], r_2[T_j^2], X_2[k-1])$ obtained from the MPCM procedure described in Lemma 1.

Proof. Let us first find the switching probability P_i. Since the time duration for UAV i to maintain its current maneuver $\tau_i[T_j^i]$ follows exponential distribution as described in Equation (2.1), P_i can be approximated from its exponential distribution as

$$P_i = \lambda_i \delta. \tag{2.17}$$

With the switching probability and the defined system mapping $G_2(v_2[T_j^2], r_2[T_j^2], X_2[k-1])$, Equation (2.14) can be further written as

$$\begin{aligned}&E(X_2[k]|X_2[k-1])\\&=P_2 E[G_2(v_2[T_j^2], r_2[T_j^2], X_2[k-1])]\\&\quad + (1-P_2) f(v_2[k-1], r_2[k-1], X_2[k-1])\end{aligned} \tag{2.18}$$

From Lemma 1, one has $E[G_2(v_2[T_j^2], r_2[T_j^2], X_2[k-1])] = E[G_2'(v_2[T_j^2], r_2[T_j^2], X_2[k-1])]$. As such, Theorem 1 can be derived naturally by combining Equation (2.18) and Lemma 1. □

Theorem 1 provides an approach to estimate the expected system state for random switching systems with computational efficiency, which we will use for the state estimation of the remove UAV from the measurement $Z_{G,2}[k]$. In particular, we integrate MPCM and UKF for a 5-step state estimation procedure. *Steps 1* and *2* select initial conditions and MPCM points to initialize *Steps 3-5*; *Step 3* finds the state estimator when the switching behavior $s[k-1]=0$; *Step 4* finds the state estimator when the switching behavior $s[k-1]=1$; *Step 5* finally finds the expected state by integrating the two estimators found in *Steps 3* and *4*.

Step 1: Initialize. Select initial conditions $\hat{X}_2[0]$ and $P[0]$ to initialize the system.

Step 2: Select MPCM points. n_1n_2 MPCM simulation point pairs are selected for the random variables $v_2[T_j^2]$ and $r_2[T_j^2]$ according to the MPCM procedure [37, Section II][38]. Denote the selected MPCM point pairs as $(\mathcal{V}_{j_1}[T_j^2], \mathcal{R}_{j_2}[T_j^2])$, where $j_1 \in \{0, ..., n_1-1\}$ and $j_2 \in \{0, ..., n_2-1\}$).

Step 3: Estimate the state when $\mathbf{s[k-1]=0}$***.*** When $s[k-1]=0$, the remote UAV does not change its maneuver, and hence the conditional expected current state $E(X_2[k]|\hat{X}_2[k-1], s[k-1]=0, Z_{G,2}[k])$ can be estimated using UKF through the following four steps: (a) select sigma points from $\hat{X}_2[k-1]$; (b) predict system state by instantiating the sigma points through the system dynamics $f_2(.)$; (c) select new sigma points from the predicted state, and predict measurement by instantiating the sigma points through the measurement model $h_{G,2}$; (d) update the Kalman Gain and find the expected state $E(X_2[k]|\hat{X}_2[k-1], Z_{G,2}[k], s[k-1]=0)$ and covariance $E(P[k]|P[k-1], Z_{G,2}[k], s[k-1]=0)$. Please refer to [39] for the detailed procedure.

Step 4: Estimate the state when $\mathbf{s[k-1]=1}$***.*** When $s[k-1]=1$, the remote UAV changes its maneuver according to the random variables $v_2[T_j^2]$ and $r_2[T_j^2]$. With the MPCM points selected in *Step* 2 $(\mathcal{V}_{j_1}[T_j^2], \mathcal{R}_{j_2}[T_j^2])$, the expected state $E(X_2[k]|\hat{X}_2[k-1], Z_{G,2}[k], s[k-1]=1)$ and covariance $E(P[k]|P[k-1], Z_{G,2}[k], s[k-1]=1)$ can be estimated using the following two sub-steps.

(a). Estimate system state at each selected MPCM point. The system state is estimated at each selected MPCM point $(\mathcal{V}_{j_1}[T_j^2], \mathcal{R}_{j_2}[T_j^2])$ by conducting the UKF procedures shown in *Step 3, (a)-(d).* Denote the estimated state from UKF at each MPCM point as $\hat{X}_{j_1j_2}[k|k-1, Z_{G,2}[k], s[k-1]=1]$ with the covariance $P_{j_1j_2}[k|k-1, Z_{G,2}[k], s[k-1]=1]$.

(b). Estimate the expected state using MPCM. With the selected MPCM points and the derived estimated states at these points, a reduced polynomial mapping from $(v_2[T_j^2], r_2[T_j^2])$ to the estimated state can be obtained according to Lemma 1

$$G'_X(v_2[T_j^2], r_2[T_j^2]) = \sum_{j_1=0}^{n_1-1}\sum_{j_2=0}^{n_2-1} \Omega_{X_{j_1j_2}} v_2^{j_1}[T_j^2] r_2^{j_2}[T_j^2],$$

where $(v_2[T_j^2], r_2[T_j^2])$ and $G'_X(v_2[T_j^2], r_2[T_j^2])$ take the values of $(\mathcal{V}_{j_1}[T_j^2], \mathcal{R}_{j_2}[T_j^2])$ and $\hat{X}_{j_ij_2}[k|k-1, Z_{G,2}[k], s[k-1]=1]$, respectively. The coefficients $\Omega_{X_{j_1j_2}}$ can be determined using the least squares method.

Similarly, the reduced-order polynomial mapping from $(v_2[T_j^2], r_2[T_j^2])$ to the error covariance can be obtained as

$$G'_P(v_2[T_j^2], r_2[T_j^2]) = \sum_{j_1=0}^{n_1-1}\sum_{j_2=0}^{n_2-1} \Omega_{P_{j_1j_2}} v_2^{j_1}[T_j^2] r_2^{j_2}[T_j^2],$$

where $(v_2[T_j^2], r_2[T_j^2])$ and $G'_P(v_2[T_j^2], r_2[T_j^2])$ take the values of $(\mathcal{V}_{j_1}[T_j^2], \mathcal{R}_{j_2}[T_j^2])$ and $P_{j_1j_2}[k|k-1, Z_{G,2}[k], s[k-1]=1]$, respectively. $\Omega_{P_{j_1j_2}}$ are the least squares coefficients.

With the polynomial mappings, the expected state and covariance can be derived according to MPCM's mean calculation procedure [37]: $E(X_2[k]|\hat{X}_2[k-1], Z_{G,2}[k], s[k-1]=1) = E[G'_X(v_2[T_j^2], r_2[T_j^2])]$, and $E(P[k]|P[k-1], Z_{G,2}[k], s[k-1]=1) = E[G'_P(v_2[T_j^2], r_2[T_j^2])]$.

Step 5: Estimate the expected system state. The estimated state and covariance are derived according to Theorem 1.

$$
\begin{aligned}
&E(X_2[k]|\hat{X}_2[k-1], Z_{G,2}[k]) \\
=&P_2 E(X_2[k]|\hat{X}_2[k-1], Z_{G,2}[k], s[k-1]=1]) \\
&+ (1-P_2)E(X_2[k]|\hat{X}_2[k-1], Z_{G,2}[k], s[k-1]=0]), \\
&E(P[k]|P[k-1], Z_{G,2}[k]) \\
=&P_2 E(P[k]|P[k-1], Z_{G,2}[k], s[k-1]=1]) \\
&+ (1-P_2)E(P[k]|P[k-1], Z_{G,2}[k], s[k-1]=0]).
\end{aligned}
$$

As such, the estimate of $X_2[k]$ is $\hat{X}_2[k] = E(X_2[k]|\hat{X}_2[k-1], Z_{G,2}[k])$, and the expected error covariance is $P[k] = E(P[k]|P[k-1], Z_{G,2}[k])$.

2.3.1.2 Adaptive Optimal Control

An on-line adaptive optimal controller is designed to maximize the expected value function (2.11) with the estimated system state.

As the uncertain parameters are independent from the states, the value function can be further re-written as

$$
\begin{aligned}
V(X[k]) =& E[\sum_{l=k}^{k+N} \alpha^{l-k} Z_R[l](X[l], U[k])] \\
=&E[Z_R[k](X[k], U[k]) + \sum_{l=k+1}^{k+N} \alpha^{l-k} Z_R[l](X[l], U[k])].
\end{aligned}
\tag{2.18}
$$

The above equation can be solved backward-in-time using dynamic programming, or forward-in-time using RL [27,28]. Here we use RL, in particular, the policy iteration method, to find the optimal control policy by iteratively conducting the two steps: policy evaluation and policy improvement. The policy evaluation step is designed to solve the value function $V(X[k])$ using Equation (2.18), given the current control policy. The policy improvement step is designed to find the best control policy to minimize the value function [27, 28]. The two steps are conducted iteratively until convergence.

Policy Evaluation

$$\begin{aligned} V_{j+1}(X[k]) =& E[Z_R[k](X[k], U[k]) \\ &+ \sum_{l=k+1}^{k+N} \alpha^{l-k} Z_{j,R}[l](X[l], U[k])] \end{aligned} \tag{2.19}$$

Policy Improvement

$$\begin{aligned} U_{j+1}(X[k]) =& \arg\max_{U_j[k]} E[Z_R[k](X[k], U[k]) \\ &+ \sum_{l=k+1}^{k+N} \alpha^{l-k} Z_{j+1,R}[l](X[l], U[k])] \end{aligned} \tag{2.20}$$

where j is the iteration step index.

Note that Equation (2.19) involves three unknown parameters for the environment-specific RSSI model ($G_{t|dB}^{max}$, $G_{t|dBm}^{min}$, and θ_{env}), which need to be learned. In particular, for each iteration j, three time steps (k, $k+1$ and $k+2$) are needed to come up with three equations to iteratively solve for the three parameters. To calculate the value function $V_{j+1}(X[k])$ at each time step, $E[\sum_{l=k+1}^{k+N} \alpha^{l-k} Z_{j,R}[l](X[l], U[k])]$ is approximated by the output mean of a system mapping using MPCM, $G_V(v_2[T_j^2], r_2[T_j^2]) = E[\sum_{l=k+1}^{k+N} \alpha^{l-k} Z_{j,R}[l](X[l], U[k])]$.

2.3.2 Using the Learned RSSI Model in both GPS-available and GPS-denied Environments

With the learned RSSI model, the optimal solution can then be obtained in both GPS-available and GPS-denied environments. In a GPS-denied environment, the RSSI is the only measurement. In this case, the optimal control solution can be found following a similar procedure as shown in Chapter 2.3.1, by replacing $Z_{G,2}[k]$

and $h_{G,2}$ with $Z_R[k]$ and h_R. In the GPS-available environment, GPS and RSSI measurements can be fused to estimate the system states, using a fuzzy-logic based fusion algorithm [14] to improve the reliability. The details are omitted here due to the limited space.

2.4 Simulation Studies

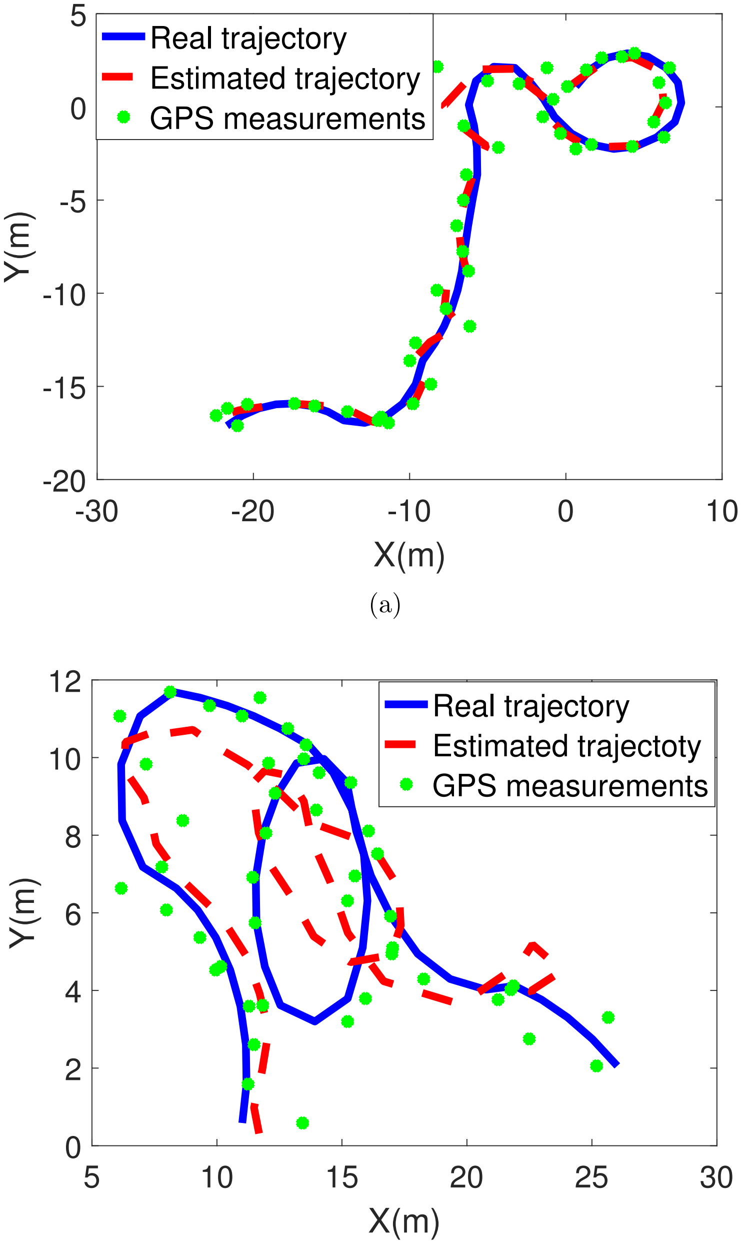
In this section, we conduct simulation studies to illustrate and validate the results and algorithms developed in this chapter. Two UAVs move in a 2-D airspace following the ST RMM independently. Two directional antennas of the same type are mounted on the two UAVs respectively. The total simulation time is $T = 45s$, with the sampling period $\delta = 1s$.

We first simulate the case when the GPS is available but the RSSI model is unknown. Gaussian noise is added to the GPS measurements. Estimation for UAV 1 is based on UKF with known maneuver ($v_1[k]$ and $r_1[k]$), while the estimation for UAV 2 is based on the integration of UKF and MPCM as described in Chapter 2.3.1 with unknown $v_2[k]$ and $r_2[k]$. Figures 2.2(a) and 2.2(b) show the trajectories of UAV 1 and UAV 2 respectively.

It can be seen from the figures that 1) the estimated trajectories for UAV 1 and 2 are both close to their real trajectories, indicating that the proposed state estimation algorithm performs well in both known and unknown maneuver cases; 2) compared with UAV 2, the estimated trajectory of UAV 1 is closer to its real trajectory as expected, indicating that the state estimation algorithm with known maneuver guarantees a better performance.

With the estimated states, we simulate the RL-based stochastic optimal control algorithm. To simulate the long-distance communication scenario, the minimum received signal strength is assumed to be 0, and in this case, the directional antennas' minimum gain ($G_{t|dBi}^{min}$) can be calculated accordingly. Figures 2.3(a) and 2.3(b) show the learned environment-specific antennas' maximum gain ($G_{t|dBi}^{max}$) and the shift angle caused by the environment ($\theta_{t_{env}}$) respectively. Gaussian noises are added to the RSSI measurements. As shown in the figures, the learned parameters are very close to their true values, which indicates the effectiveness of the learning algorithm. The derived optimal heading angles are very close to the true optimal angles, indicating the good performance of the proposed RL-based stochastic optimal control algorithm.

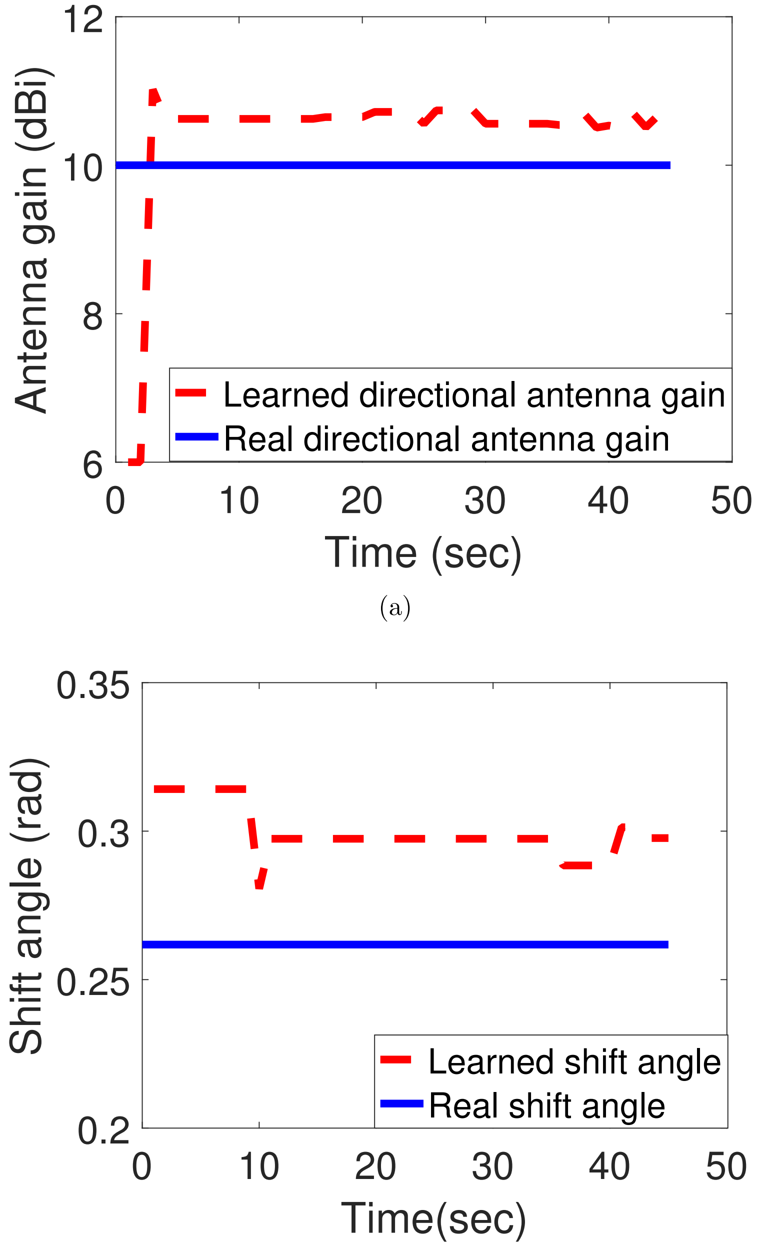
With the learned RSSI model, we simulate the proposed stochastic optimal control algorithm in GPS-denied and GPS-available environments respectively. Figures 2.4 and 2.5 show the estimation and control performances in GPS-denied and GPS-available cases respectively. In the GPS-denied environment, RSSI is the only measurement. We here allow the angle velocity $\omega_2[k]$ to be transmitted through the air-to-air communication channel. It can be seen from the figures that: 1) the estimated trajectories and derived heading angles are very close to their true trajectories and real optimal heading angles in both cases, indicating that the proposed solutions work well in both GPS-available and GPS-denied environments; 2) the angle errors in the GPS-available case are much smaller than those in the GPS-denied case, indicating that the fusion of the GPS and RSSI promises a better performance.



(a)

(b)

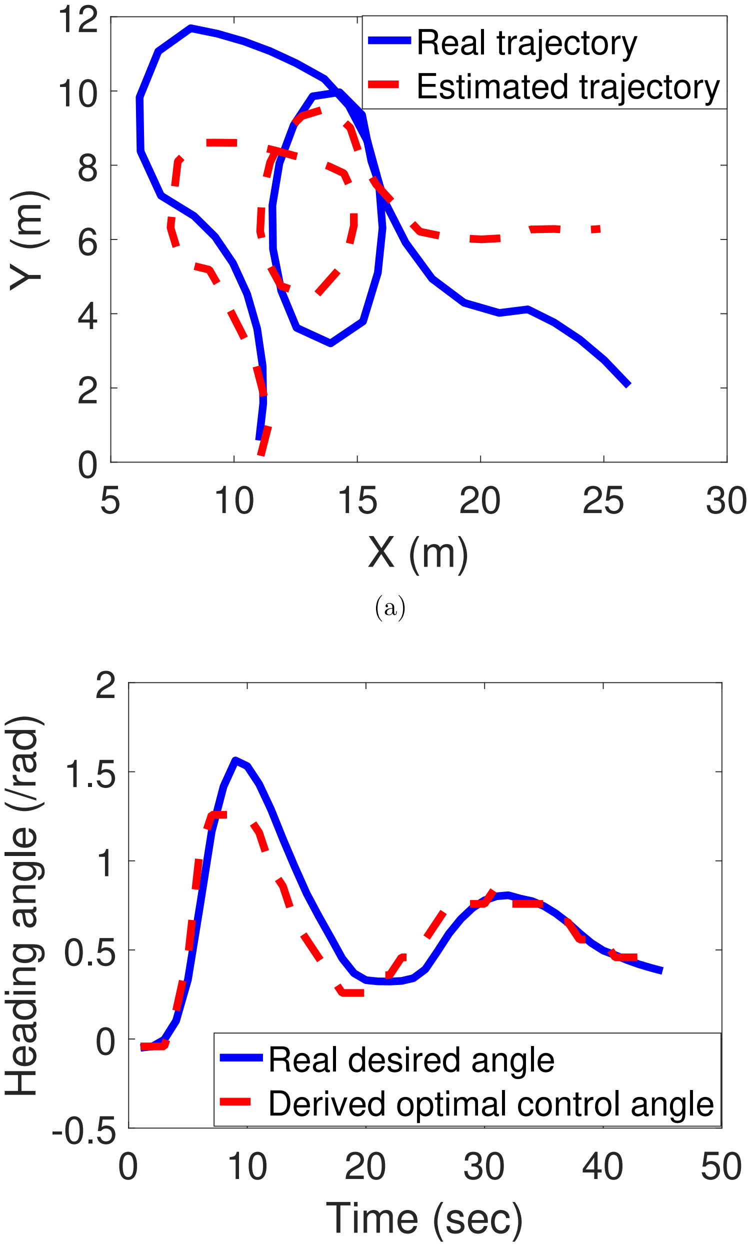
Figure 2.2. (a) Trajectories of UAV 1, and (b) Trajectories of UAV 2. The blue solid curves are real trajectories, red dotted curves are estimated trajectories, and green dots are GPS measurements.



(a)

(b)

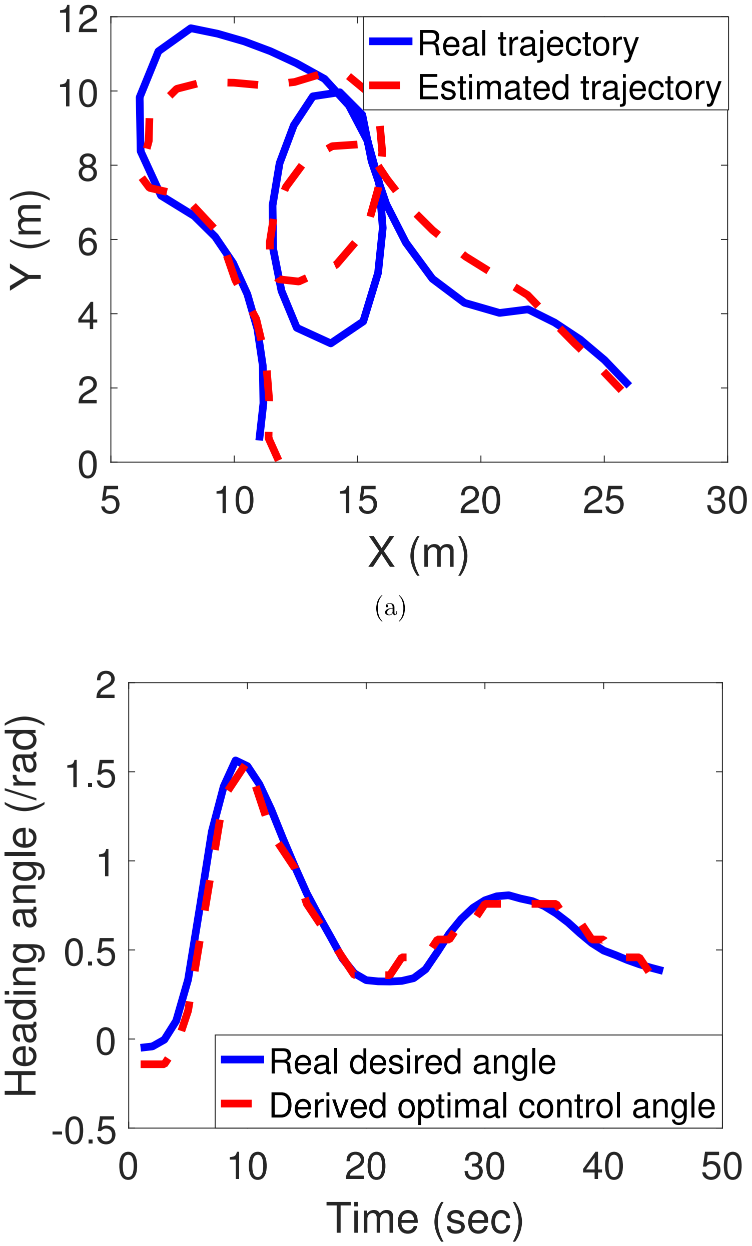
Figure 2.3. Learned environment-specific (a) maximum directional antenna gain ($G_{t|dBm}^{max}$), and (b) shift angle (θ_{env}) in the RSSI model. The blue solid lines are the real parameter values, and the red dotted curves are the learned parameters.



(a)

(b)

Figure 2.4. (a) Performances of the developed (a) estimation algorithm, and (b) controller algorithm, in GPS-denied environment. The blue solid curves are the real trajectories (or real desired angles), and the red dotted curves are the estimated trajectories (or derived control angles respectively).



(a)

(b)

Figure 2.5. (a) Performances of the developed (a) estimation algorithm, and (b) controller algorithm, in GPS-available environment. The blue solid curves are the real trajectories (or real desired angles), and the red dotted curves are the estimated trajectories (or derived control angles respectively).

CHAPTER 3

The Design and Implementation of Aerial Communication Using Directional Antennas: Learning Control in Unknown Communication Environments

3.1 Introduction

UAV networking extends the capability of a single UAV to multiple UAVs, and is foreseen to have broad uses in a wide spectrum of commercial UAV applications, such as on-demand emergency [40, 41], surveillance [10], reconnaissance [42], and remote sensing [43]. Several communication schemes for low-altitude UAV platforms have been developed, such as omni-directional networking [44, 45] and cellular-based communication [46]. In the past few years, we developed several solutions using UAV-carried directional antennas to realize the on-demand broadband and long-distance ACDA UAV-to-UAV communication system[11–13, 40, 41]. The performance of the ACDA system was verified using both simulation studies and field tests.

In order to improve the UAV-to-UAV communication quality, communication performance indicators can be utilized to assist with the antenna controller design. Widely used wireless communication quality indicators include the channel capacity [47], bit error rate (BER) [48], signal-to-noise ratio (SNR) [49], and received signal strength indicator [50, 51]. As studied in [47–49], all of these indicators are related to the transmitting and receiving signal powers. Our special interest here is the RSSI, which measures the signal power received from the transmitting antenna, and can be directly obtained from the received signal without additional measurement equipment. With the same transmitting signal power, a stronger RSSI leads to a higher SNR, and in turn a lower BER [48] and a larger channel capacity [47]. As such, we adopt the RSSI as the communication performance indicator to quantify the performance of UAV-to-UAV wireless communication channel in ACDA.

In [14], the RSSI was used as an additional measurement signal to compensate GPS signals for the automatic alignment of directional antennas, and a proportional-integral (PI) controller was used for antenna control. This controller is designed based on two assumptions: 1) the RSSI model is known ahead of time, and 2) the strongest RSSI is achieved when two directional antennas point towards each other. These assumptions may not hold in an imperfect communication environment, considering the existence of reflection, refraction and absorption by buildings, obstacles, and interference sources, which complicate the communication channel model. In this chapter, we aim to address this issue and develop an autonomous antenna heading alignment solution that optimizes the antennas' headings to achieve maximal RSSI in real imperfect environments, with the communication channel model unknown in advance. In particular, we develop a Reinforcement Learning-based online optimal controller that learns the environment-specific communication channel model in real time.

The contributions of this chapter are summarized as follows.

- *Combined Communication Channel for Application, Control and Command Data.* Application, control, and command data share the same directional antenna-equipped UAV-to-UAV channel. This networking design is very different from the previous versions which separate the high-rate directional antenna-equipped application channel and the low-rate omni-directional antenna-equipped control and command channel. This solution improves the robustness of the ACDA system, as the omni-directional antenna-equipped control and command channel is sensitive to interferences in long-distance set-ups.

- *RL-based Directional Antenna Control that Maximizes Communication Performance in Unknown Communication Environments.* A partial solution presented in Chapter 2 is designed and implemented. In particular, an RL-based online optimal controller is developed to find the optimal heading angles that maximize the RSSI of directional antennas in an imperfect environment. This solution is also coupled with an algorithm that learns an environment-specific communication channel model in an unknown communication environment. This antenna control solution maximizes the communication performance in real settings, as it does not require a known and perfect communication channel model assumed in the previous versions of ACDA.
- *Integrated Design and Implementation of a High-performance ACDA System.* We develop an ACDA system with an integrated design and implementation of the communication, control, and computing components. The implementation adopts the Robot Operating System (ROS) to facilitate communication and data transmissions. The computing functionality is implemented using BeagleBone Black, a compact low-power single-board computer suitable for UAV applications [52]. Compared to Arduino adopted in our previous ACDA systems, BeagleBone Black features a powerful CPU, multiprocessing capability, and ROS support.
- *Application to Emergency Communication Provision with the Consideration of Practical Needs.* We apply the ACDA system in emergency scenarios, by providing an on-demand broadband communication channel to transmit both optical and infrared monitoring videos of the disaster zone over a long distance to the emergency management center [40, 41]. Special considerations of emergency practices are considered in the design and implementation of the ACDA system. In particular, we develop an interface using ROS and Qt, a cross-platform

application framework [53]. The interface displays at ground stations the status information of the entire system, such as GPS, antenna headings, and RSSI for system diagnosis. It also displays video streams captured by the remote UAV. In addition, a water-resistant enclosure is designed to protect the ACDA system from adverse environmental conditions for practical use.

The rest of this chapter is organized as follows. In Chapter 3.2, we describe the ACDA system design, including both hardware and software structures. In Chapter 3.3, we elaborate the RL-based on-line directional antenna control solution. Chapter 3.4 verifies the directional antenna control solution and evaluates the performance of the ACDA system using simulation studies, field tests and disaster drills.

3.2 The ACDA System Design

In this section, we first provide an overview of the ACDA system, and then provide the detailed description and analysis for each component, including the UAV platform, communication, computing, antenna heading control, cameras, ROS, interface, and water-resistant design.

3.2.1 Overview

The objective of the ACDA system is to provide a robust and cost-effective long-distance and broadband UAV-to-UAV communication channel for on-demand emergency communication, and meanwhile, to offer a user-friendly interface that simplifies the operation and diagnosis procedures. The ACDA system is composed of a pair of integrated subsystems on both the local side and the remote side. The subsystem on each side includes a laptop, a UAV platform, and the onboard communication, computing, and control components (see Figure 3.1).

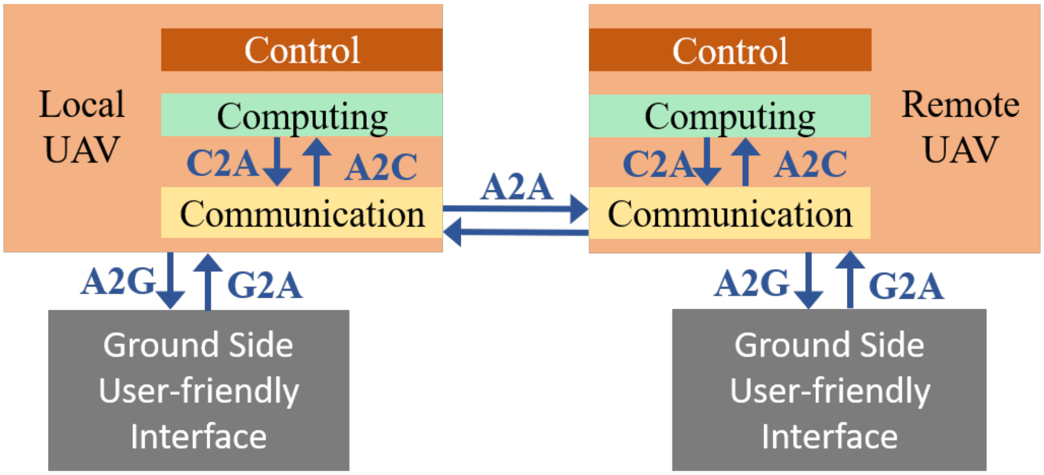


Figure 3.1. Overview of the ACDA system.

The role of the communication component is to connect modules of the ACDA system through wired and wireless links. Three types of wireless communication links in the ACDA system include the Air to Ground (A2G) / Ground to Air (G2A), Air to Air (A2A), and Air to Computing module (A2C) / Computing module to Air (C2A) links. We adopt multiple communication technologies to avoid the mutual interference. The role of the computing component is to collect data from sensors and the other components, process the computing and learning tasks, and then output the motor control signal to the control component. The role of the control component is to drive the motor that rotates the directional antennas for autonomous alignment and form a robust long-distance communication channel.

Figure 3.2 shows the modules of each component and their connections of the subsystem on one side. The communication component includes a directional antenna, a Wi-Fi router, and a Wi-Fi adapter. The computing component is realized on a microprocessor. The control component contains a rotating motor, a tunable plate, a motor driver, and a compass module. The user interface installed on the

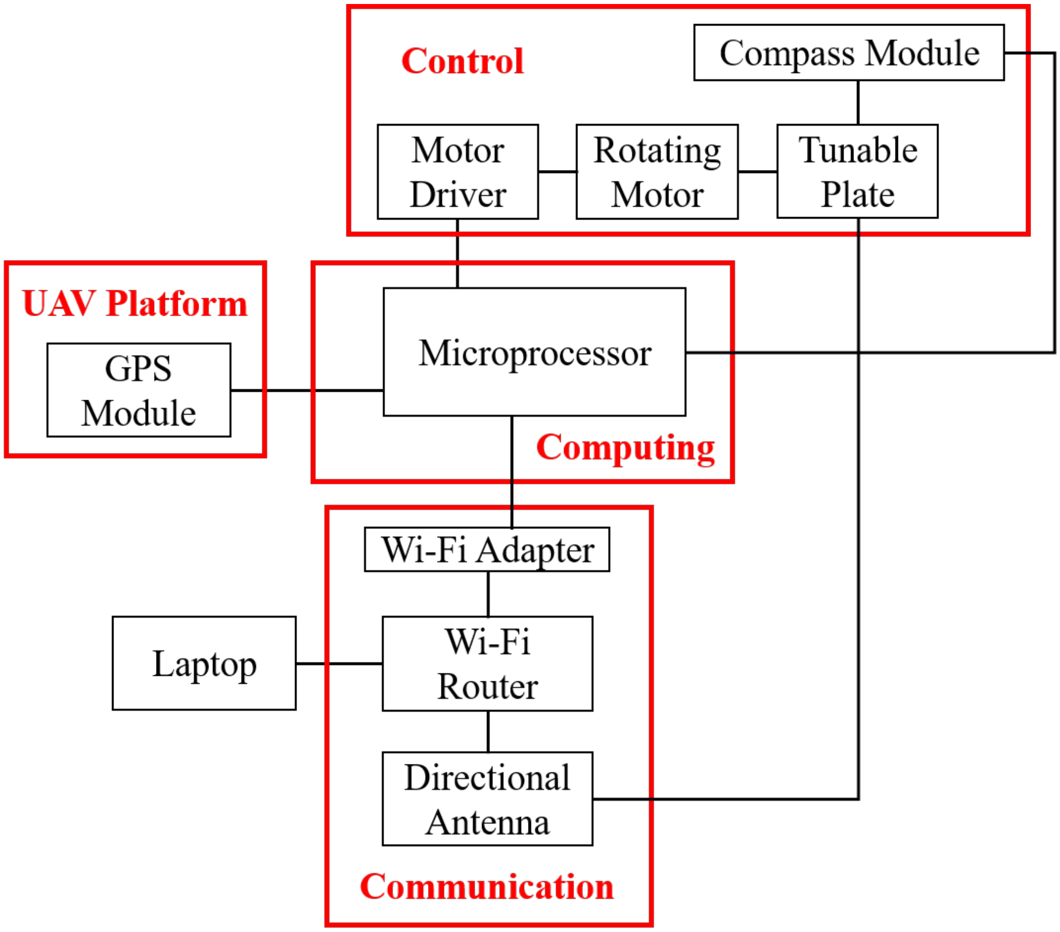


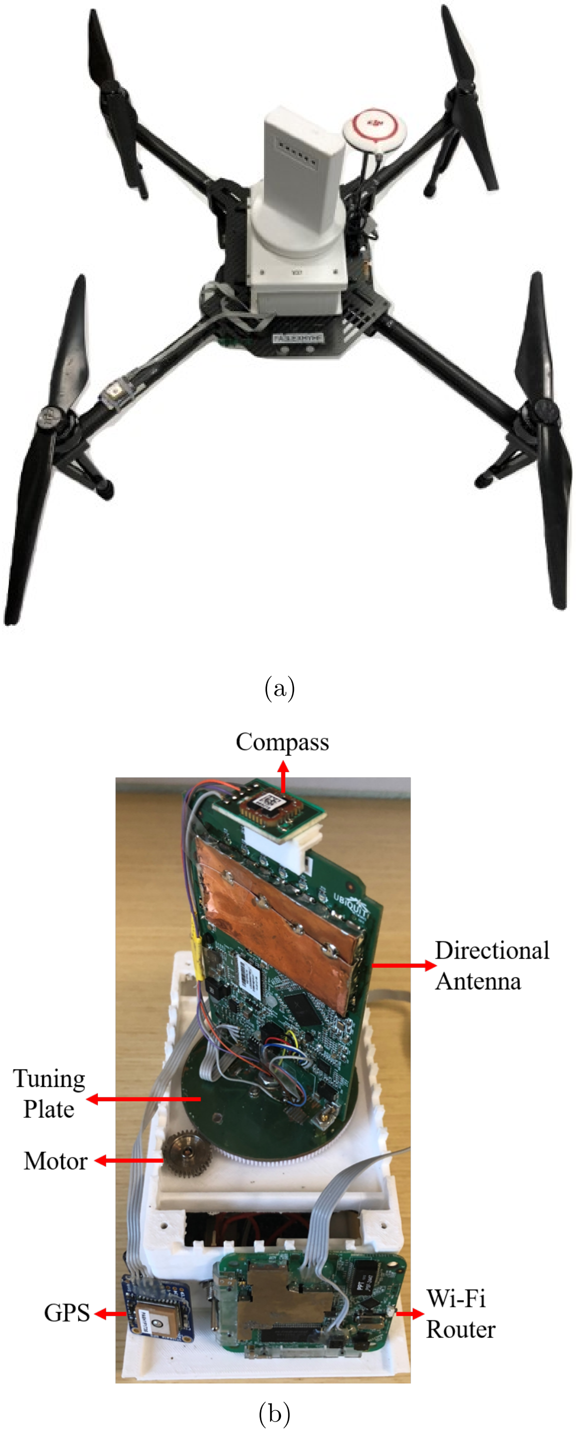
Figure 3.2. The components of the subsystems at one side of the ACDA system .

laptop is connected to the communication component through a Wi-Fi router. The GPS module on the UAV platform is connected to the microprocessor.

To facilitate practical use of the ACDA system in emergency scenarios, a water-resistant enclosure is designed to protect the core components. Please refer to Figure 3.3 for photographic illustration of the ACDA prototype.

3.2.2 The UAV Platform

The UAV platform carries the ACDA system and also performs various flight missions. Considering the performance metrics such as payload, flight time, expand-


(a)

(b)

Figure 3.3. (a) A picture of the implemented ACDA prototype, and (b) Some internal modules of the ACDA system.

ability, stability, and operability, we select DJI Matrice 100, a quad-copter, as our UAV platform (see Figure 3.3). An alternative candidate of the UAV platform is Tarot 650, which also has good performance as we analyzed in [13]. The features of the DJI Matrice 100 that match our application needs are summarized as follows [54].

(1) *Payload and Flight Time*: Matrice 100 is made of durable and lightweight carbon fiber. The maximum takeoff weight of DJI Matrice 100 is 3.6kg, larger than the total weight of the system (3.13kg), including the platform, the ACDA prototype, and a TB48D battery. The TB48D, a LiPo 6S battery, can support a flight time of about 18 minutes based on our flight tests.

(2) *Expandability*: The expansion bay on the top of Matrice 100 nicely installs the onboard components. In addition, Matrice 100 provides universal power and communication ports, including dual parallel CAN ports and dual UART ports. Matrice 100 is also programmable using the DJI Software Development Kit (SDK), which supports Linux, ROS, Qt and STM32 development environments.

(3) *Stability*: Matrice 100 uses the N1 autopilot, which maintains a stable flight including hovering, under up to 10m/s wind disturbances. Each of the arms incorporates a dampening component that eliminates the vibrations generated from the motors, to improve the stability of the system. In addition, the landing pads are installed at the base of each arm to protect Matrice 100 during landing.

(4) *Operability*: Like other commercial UAVs, the DJI Matrice 100 is ready to fly once open box. The remote controller adopts a simple design. All parameters can be monitored and tuned via a mobile app called "DJI GO". This app supports a live monitoring of control parameters and real-time videos from any connected smart phones. Some critical information including battery time and GPS signal strength are displayed.

We install a GPS module, Adafruit ultimate GPS breakout version 3 on the DJI Matrice 100 frame to support the ACDA system. It is a high-quality GPS module that can track up to 22 satellites on 66 channels with a high-sensitivity, high-speed receiver and a built-in antenna. In addition, it has the data-logging capability that can record up to 16 hours of data using a FLASH memory.

3.2.3 The Communication Component

The communication component is crucial for the ACDA system, which includes both intra- and inter- device links. The communication links in the ACDA system include the wired links between the directional antenna and the Wi-Fi router, and the wireless links for the A2A, A2G, G2A, A2C, and C2A communications (see Figure 3.4).

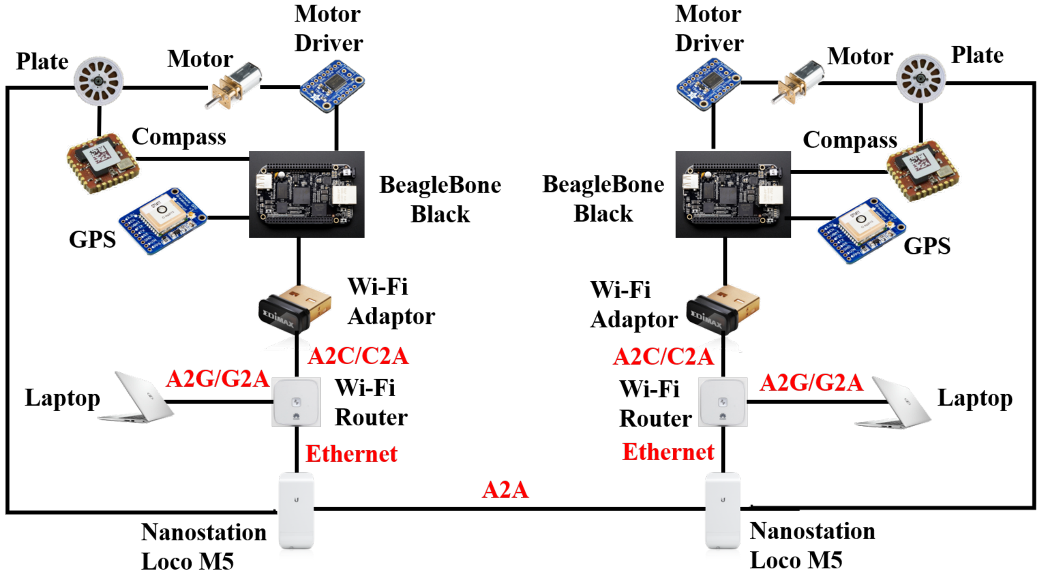


Figure 3.4. Illustration of the connections among different components of the ACDA system, with a highlight on the communication links.

(1) *A2A Link*: We select Ubiquiti Nanostation Loco M5 [35] as the directional antenna unit to establish the A2A communication link, based on an evaluation of the performance, scalability, cost, size and weight. Utilizing the 2×2 multiple-input and multiple-output (MIMO) technology, Loco M5 is a small form factor unit operating at the 5 GHz band with default antenna gain of 13 dBi. In addition, it adopts an improved time division multiple access (TDMA) technology, enabling the unit to connect multiple clients with low latency. The data rate is up to 150 Mbps and the maximum transmission distance is up to 10 km. Three modes can be selected according to application scenarios, including access point (AP) mode, client mode and wireless distribution system (WDS) mode. In our design, both the application data, and the command and control signals are transmitted between the local and remote subsystems through this A2A link.

(2) *Wired Link*: The Wi-Fi router, Huawei WS323 [55], is connected to Loco M5 through a Ethernet cable. Huawei WS323 supports both 2.4 GHz and 5 GHz bands and complies with the IEEE 802.11n standard. It adopts a 2×2 MIMO and delivers a wireless transmission rate of up to 300 Mbps while simultaneously providing the multi-device access capabilities.

(3) *A2G/G2A Links*: Data from both the local and remote UAVs are transmitted to the ground in real time using the A2G link. We use a laptop on the ground on each side to receive the application data and diagnosis information of the system, which are displayed on a user-friendly interface (to be described in Chapter 3.2.8). The laptop is connected to Huawei WS323 through the wireless local area network (WLAN). The G2A link is a reverse process of A2G. By employing the interface, user control and command signals can be sent to the Huawei WS323 located on the corresponding UAV, and then Huawei WS323 relays the information to their destinations in the network.

(4) *A2C/C2A Links*: Data are transmitted to the microprocessor for processing through the A2C link. Beaglebone Black, the microprocessor, is responsible for all the computing and learning tasks based on data received through a Wi-Fi adapter from other components in the network. The C2A Link is a reverse process of the A2C link. The microprocessor sends processing results to its destinations in the network through the Wi-Fi adapter.

Other than the aforementioned wireless links designed for the ACDA system, the operation of the ACDA system also relies on several other wireless links. Altogether, there are five types of wireless links required for the ACDA system: 1). GPS signal link (1575MHz) which receives the location information from GPS satellites, 2). flight control signal link (2.4GHz) which transmits control commands of the remote controller to the corresponding UAV, 3). A2A link (5GHz) which connects the local and remote subsystems of the ACDA system, 4). A2G/G2A link (2.4GHz/5GHz) connects the laptops to the ACDA network, and 5). A2C/C2A link (2.4GHz) which connects the microprocessors to the ACDA network.

We here analyze the aforementioned wireless links and provide a configuration strategy to effectively utilize different frequency bands to avoid the mutual interference among the communication links. The GPS signal operates on a frequency which doesn't cause the interference with other links. Since BeagleBone Black supports only 2.4GHz, the A2C/C2A links can only operate on the 2.4GHz frequency band, and hence we configure the Huawei WS323 to operate on 2.4GHz. Because the A2C/C2A and the A2G/G2A are supported by the same Wi-Fi router, i.e., Huawei WS323, A2G/G2A link also operates on 2.4GHz. The A2G/G2A, A2C/C2G, and the flight control signal link share the 2.4GHz frequency band, which does not interfere with the A2A link. Figure 3.5 shows the frequencies and bandwidths of the wireless links in the ACDA system.

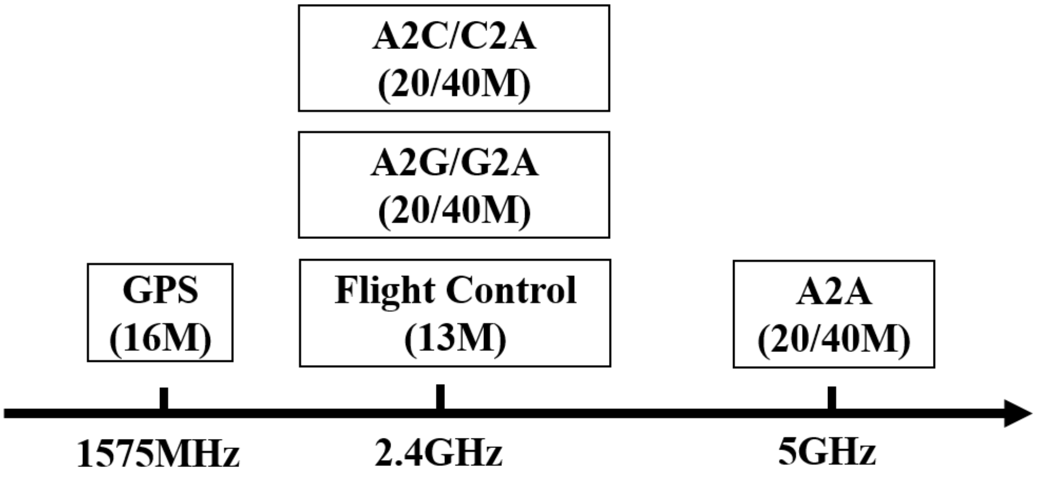


Figure 3.5. The frequencies and bandwidths of the wireless links of the ACDA system.

To avoid the mutual interference among Huawei WS323, the flight control signal link, and the communication environment, we develop an algorithm to automatically select the Wi-Fi channels for Huawei WS323 and the flight control system of DJI Matrices 100 as shown in Algorithm 1.

The communication links in the ACDA system are configured as follows.

(1). *Ubiquiti Nanostation Loco M5*: The network modes of both the local and remote Nanostation Loco M5 are set up as a bridge, which is a point-to-point link. The wireless mode of the local Nanostation Loco M5 is chosen as a client, and the remote Nanostation Loco M5 is chosen as an access point. The same WLAN service set identifier (SSID) is selected for both the local and the remote sides. The output power is set as 23 dBm. We choose the most secure option WPA2-PSK (AES), which applies the Wi-Fi protected access II algorithm with the advanced encryption standard. Channel width is set to 40MHz. The WLAN IP addresses for both sides are randomly selected in the same subnet. For example, in our system, we set the local Nanostation IP address as 192.168.33.100, and the remote IP address as 192.168.33.110.

Algorithm 1 Wi-Fi Channel Selection Algorithm

Requirement: An ACDA system placed in a given environment.

Output: Wi-Fi channels selected for both the local and remote Huawei WS323 and the flight control system of DJI Matrices 100.

Procedure:

1. Scan wireless signals and record their frequencies, channels, and RSSIs for the local subsystem.
2. Find the least crowded channel with low RSSIs and select it as the channel for local Huawei WS323.
3. Find another uncrowded channel with low RSSIs which does not overlap with local Huawei's channel, and select it as the channel for the local flight control of DJI Matrices 100.
4. Repeat the procedure 1 to 3 for the remote subsystem.

(2). *Huawei WS323*: Huawei WS323 is configured to operate on the 2.4GHz frequency band. The channel width is set to 40MHz and the WLAN communication standard of 802.11n is selected. We select different SSIDs for the local Huawei router and the remote Huawei router, which both differ from the SSID of Ubiquiti Nanostation Loco M5. The IP addresses of them should also be in the same subnet as the Nanostations. For example, we set the IP address of local Huawei router as 192.168.33.111, and the remote Huawei router as 192.168.33.121.

(3). *Wi-Fi adapter*: The IP addresses of the local and remote Wi-Fi adapters are set under the same subnet as Nanostations and Huawei routers, i.e., 192.168.33.130 for the local Wi-Fi adapter and 192.168.33.131 for the remote Wi-Fi adapter.

(4). *Laptop*: Both the local and remote laptops are set under the same subnet as other communication modules in the ACDA system, i.e., 192.168.33.21 for the local laptop and 192.168.33.122 for the remote laptop.

3.2.4 The Computing Component

The computing functionality is implemented on a microprocessor. We select BeagleBone Black, a low-cost, community-supported development platform featuring a 32-bit RISC microprocessor as the computing component [52]. It has a powerful CPU (ARM Cortex-A8) with up to 1 GHz clock time, compared to the 16 MHz clock time for the Arduino used in the previous version of ACDA. It has 512MB DDR3 RAM, 4GB 8-bit eMMC onboard flash storage, 3D graphics accelerator, NEON floating-point accelerator, and two programmable real-time unit (PRU) 32-bit microcontrollers. The BeagleBone Black supports various ports, for example, USB, Ethernet, HDMI, and 2 × 46 pin headers, making it flexible for our ACDA design. In addition, it's compatible with the Linux distributions Debian and Ubuntu, which serve as the platform for ROS, an open-source and meta-operating system. It also supports multiprocessing, which is very useful to transmit multiple sensor data streams with devices in the network. BeagleBone Black does not have a wireless card assembled, and hence we attach a portable Wi-Fi adapter to enable its communication with other components of the ACDA system.

The data processing in BeagleBone Black uses three types of data from the other components, including the GPS on the UAV platform, compass data in the control component, and RSSI retrieved from Nanostation Loco M5 through a Wi-Fi adapter. The output is the motor control signal to the control component.

3.2.5 The Antenna Heading Control Component

The antenna heading control component includes a motor driver, a rotating motor, a tunable plate, and a compass module (see Figure 3.3 (b)). The directional antenna and the compass module are placed on a tunable plate, which is controlled by a motor. The motor driver receives motor control signals from the microprocessor and outputs the pulse width modulation (PWM) signals to rotate for a specified angle.

We select Adafruit TB6612 as the motor driver. The rotating motor is a microgearmotor with a maximum speed of 130 revolutions per minute (RPM) and a gear ratio of 210:1. The tunable plate is a plastic robot gear. We use MTI-3-8A7G6T Xsens as our compass module. It has a full 3D magnetometer-enhanced attitude and heading reference system (AHRS) with an in-run compass calibration (ICC) setting, which can compensate for magnetic distortions. In addition, it provides an active heading stabilization (AHS) setting, which can significantly reduce the heading drift under magnetic disturbances. The antenna heading control algorithm to align the directional antennas for a robust A2A channel is designed in Chapter 3.3.

3.2.6 Cameras in the ACDA system

The emergency response application requires both optical and infrared monitoring videos of the remote disaster zone to be transmitted to the ground station in real-time.

On the remote side, the UAV carries a NVIDIA Jetson TX2 module that processes data from the two video sensors OV5693 and FLIR Lepton 2 (see Figure 3.6). As Jetson TX2 has a large development board, we developed a carried board of TX2 with weight 53g and size 88mm×65mm, suitable for UAV applications (see Figure 3.6a). OV5693 is a 1/4-inch, 5-megapixel image sensor, which delivers full 1080p high-definition video streams at 30 frames per second (fps). FLIR Lepton 2, a complete

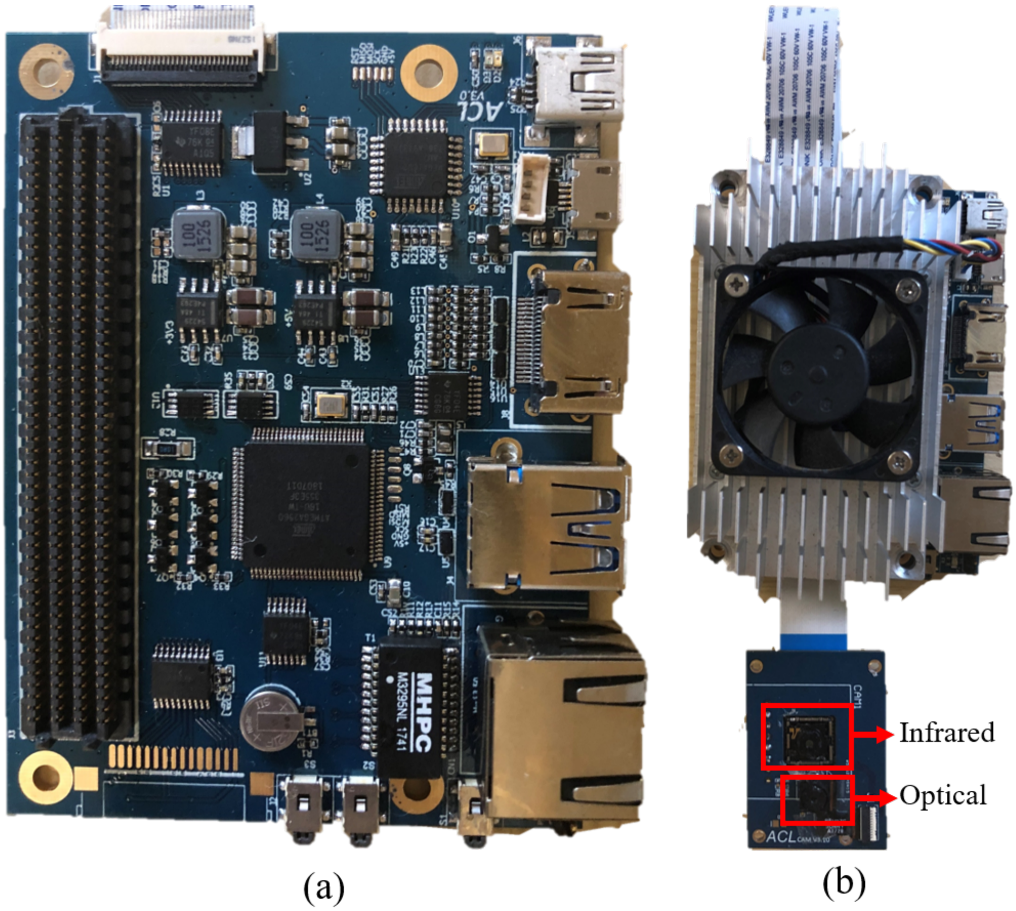


Figure 3.6. (a) Carrier board design. (b) Jetson TX2 carrier board with two cameras, i.e., the infrared camera and the optical camera.

long-wave infrared (LWIR) camera, is connected to Jetson TX2 to capture infrared radiation and output a uniform thermal image. Jetson TX2 supports Ubuntu, and hence can run ROS. Jetson TX2 is connected to the remote Wi-Fi router, i.e. Huawei WS323 through WLAN with an IP address 192.168.33.133. The local ground station receives the two video streams from the remote UAV in real time through the A2A link.

3.2.7 ROS and Information Flow

We adopt ROS to support data transmissions in the ACDA system. Here we describe ROS, and the ACDA information flow built on ROS.

ROS is a widely used middleware for developing robot applications [56]. It's a collection of tools, libraries, and conventions that enable users to build various applications by providing services such as low-level device control, message-passing between processes, and package management. It works well on traditional operating systems, such as Ubuntu in our case, and can support multi-device communication, distributed computation, and rapid testing.

ROS is composed of packages. A ROS package is a collection of files that serves for a specific purpose, generally including source files, executable files and supporting files (libraries, configuration files, dataset, etc.). ROS nodes are a set of basic independent executable units that perform various tasks. In order to share information, all nodes must communicate with each other. The primary mechanism that ROS nodes adopt for communication is fto send and receive messages. A message is a simple data structure consisting of typed fields. Messages are organized into named topics. A node shares information by publishing messages on appropriate topics, and a node receives information by subscribing to the corresponding topics. The nodes do not need to know whom they are communicating with, but only publish or subscribe messages of the topics of interest [56]. ROS allows multiple publishers and subscribers for a topic. To facilitate the communication among nodes, a ROS master provides naming and registration services to the ROS nodes. The ROS master tracks publishers and subscribers of topics and enables individual ROS nodes to locate each others.

The use of ROS nodes provides several benefits to the design and implementation of the ACDA system. First, the publish/subscribe mechanism simplifies the data transmission among multiple components in a network. Second, the system is fault tolerant as crashes are isolated to individual nodes. Third, code complexity is reduced by encapsulating small tasks into nodes and assembling them in a structured way. Forth, implementation details are hidden as the nodes expose a minimal application programming interface (API) to the rest of the system and hence can be easily modified or replaced.

Our ACDA system has the following ROS architecture. The ROS master runs on the local BeagleBone Black. It has four ROS clients, including the remote BeagleBone Black, the Jetson TX2 to support video processing, the local laptop, and the remote laptop. The ROS topics published and subscribed by the ROS master and clients are shown in Figure 3.7.

The messages in the system include GPS, RSSI, compass, and camera information. Each message here corresponds to a specific topic. The local BeagleBone Black publishes the local compass message, GPS message, RSSI message and heading angle message, and subscribes the remote GPS message and remote RSSI message to conduct the RL-based on-line directional antenna control algorithm at the local side. The remote BeagleBone Black functions in a similar way. The remote Jetson TX2 publishes two camera messages including the infrared video message and optical video message, so that the videos on the remote side can be tracked at the ground stations on the local side. The local laptop and the remote laptop subscribe all the topics in the ACDA system and display the information on a user-friendly interface to monitor the system performance and conduct various operations.

Figure 3.8 shows the ROS information flow. The local directional antenna control algorithm running on a microprocessor calculates the local antenna heading angle by using both the local and remote GPS and RSSI signals, and sends the PWM signals to the local motor driver. The information flow on the remote side is similar.

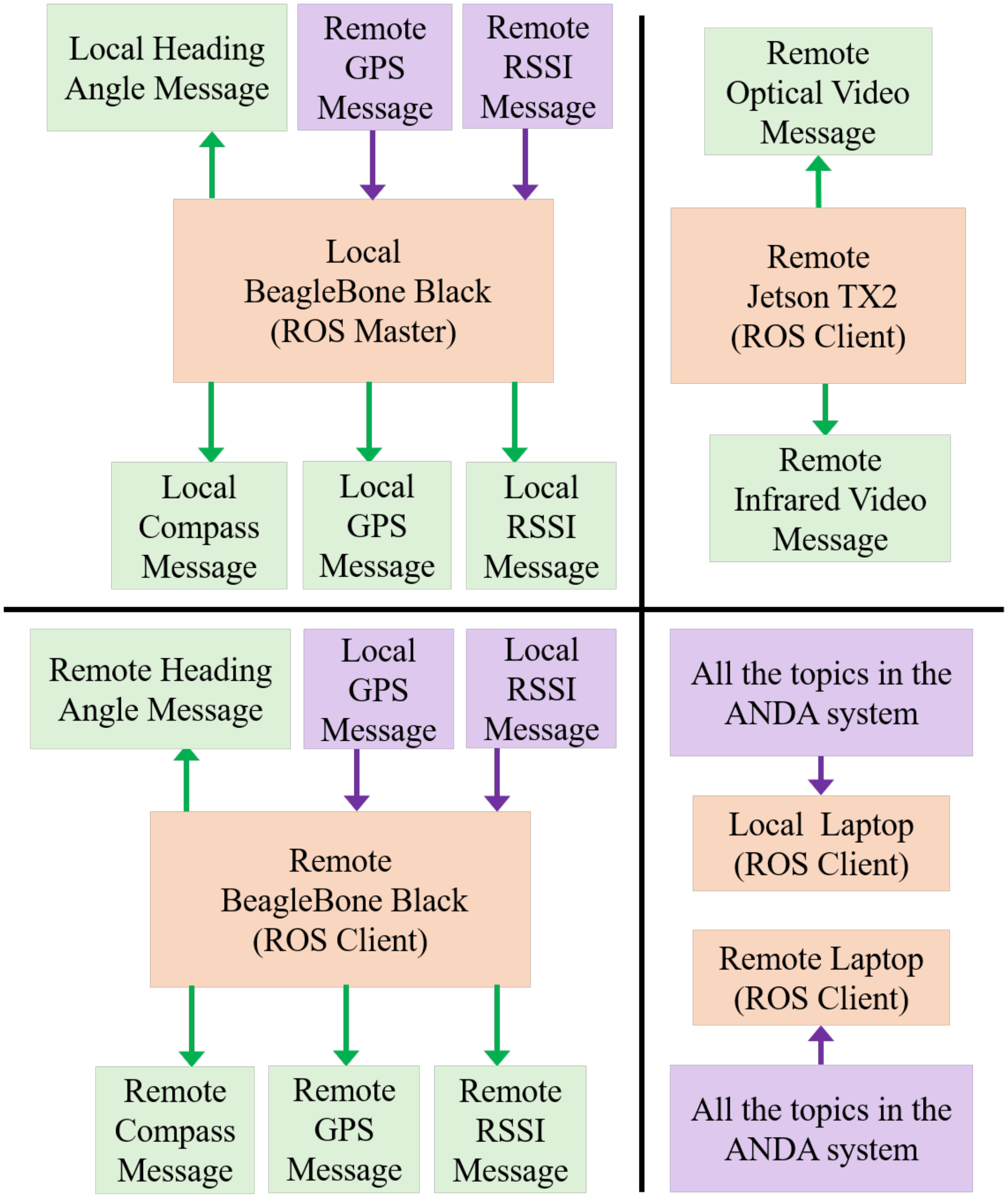

Local Heading Angle Message
Remote GPS Message
Remote RSSI Message
Local BeagleBone Black (ROS Master)
Local Compass Message
Local GPS Message
Local RSSI Message
Remote Optical Video Message
Remote Jetson TX2 (ROS Client)
Remote Infrared Video Message
Remote Heading Angle Message
Local GPS Message
Local RSSI Message
Remote BeagleBone Black (ROS Client)
Remote Compass Message
Remote GPS Message
Remote RSSI Message
All the topics in the ANDA system
Local Laptop (ROS Client)
Remote Laptop (ROS Client)
All the topics in the ANDA system

Figure 3.7. ROS architecture and topics in ACDA system. The orange blocks denote the the ROS hosts, including the ROS master and ROS clients. The green blocks denote published messages and the purple blocks denote subscribed messages.

The local and remote user interfaces receive and display all the messages in the ACDA system.

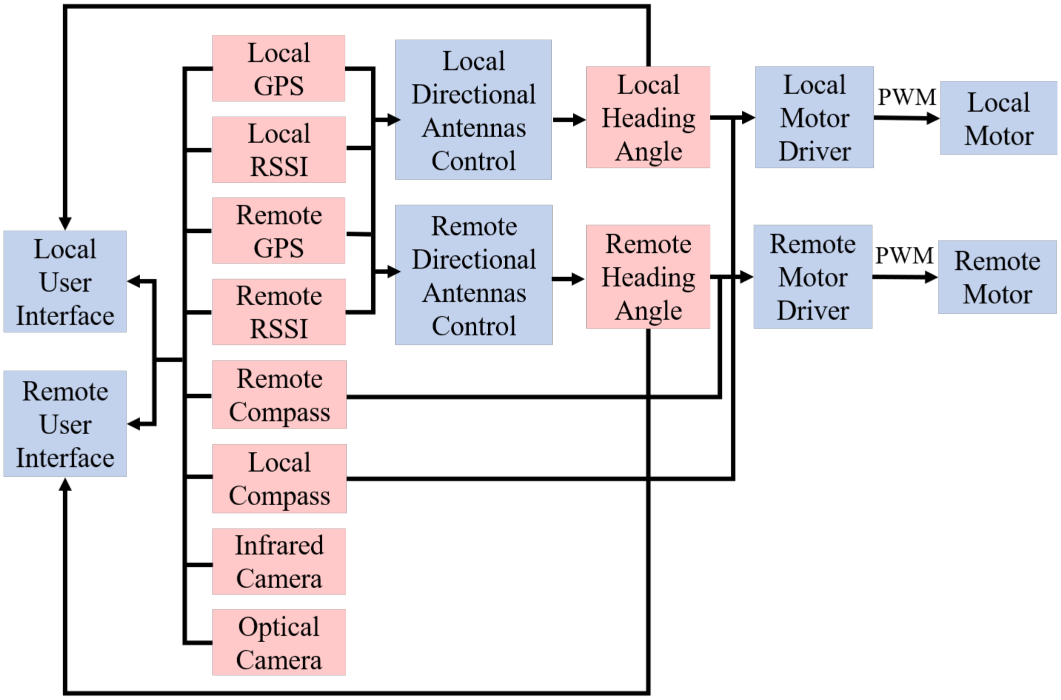
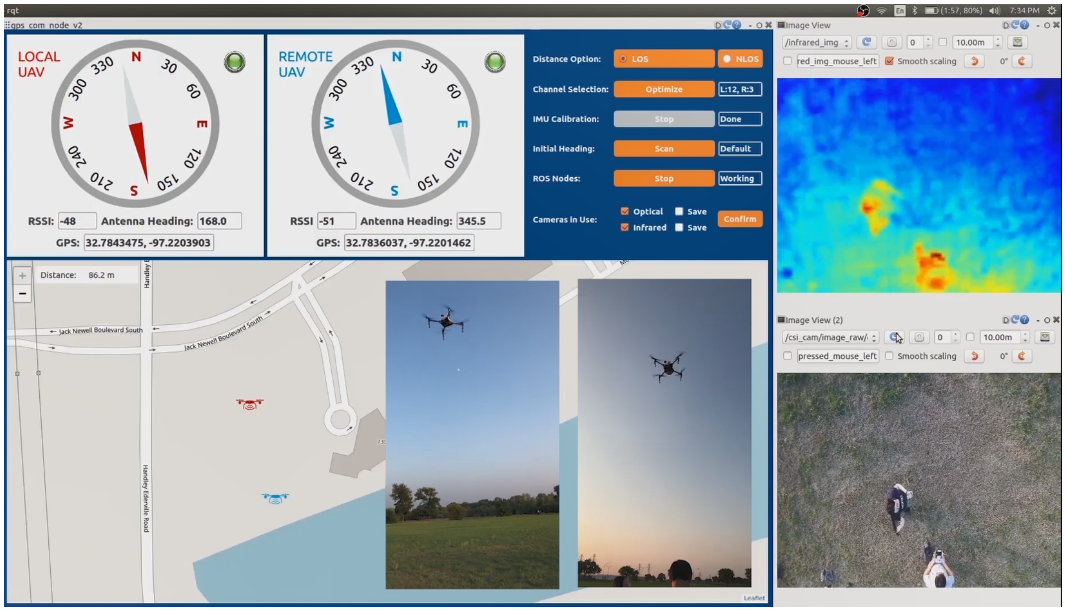


Figure 3.8. Information flow of the ACDA System. The red blocks denote messages in the system. The blue blocks denote components to which these information flow.

3.2.8 Interface Design

A user-friendly interface (see Figure 3.9) is designed to automate the configuration, operation, and diagnosis procedures. The interface displays the locations of UAVs, communication quality, antenna alignment performance, and real-time videos. In addition, control operations are included to automate the intricate configuration processes, lowering the burden for users such as emergency staff.

The graphical user interface (GUI) is developed using Qt, a cross-platform application framework and widget toolkit [53]. Qt supports all major desktop platforms and most mobile or embedded platforms. Its advantages include fast and simple programming, consistent and comprehensive APIs and libraries, and compatibility with various compilers.



RSSI: -48 Antenna Heading: 168.0
GPS: 32.7843475, -97.2203903

RSSI -51 Antenna Heading: 345.5
GPS: 32.7836037, -97.2201462

Distance Option: ● LOS ○ NLOS
Channel Selection: L:12, R:3
IMU Calibration: Done
Initial Heading: Default
ROS Nodes: Working
Cameras in Use: ☑ Optical □ Save ☑ Infrared □ Save

Distance: 86.2 m

/infrared_img 0 □ 10.00m
□ red_img_mouse_left ☑ Smooth scaling 0°

/csi_cam/image_raw/ 0 □ 10.00m
□ pressed_mouse_left □ Smooth scaling 0°

Figure 3.9. A user-friendly interface design.

The interface of the ACDA system is composed of four parts, including the message input and display, map, infrared and optical video display, and control panel. The message input and display support two functions. First, it displays information of the local and remote compasses, GPS, RSSIs and antenna headings retrieved from the onboard ACDA system. The local antenna heading angle is marked on a compass dial in red, and the remote antenna heading is marked on the compass dial in blue. The GPS, RSSI and desired antenna heading angles are displayed in text boxes. In addition, two circle indicators on the right side of the compass dials show the connection qualities of the ACDA system on the local side and the remote side respectively, where green denotes connection established and red denotes connection lost. Second, the text boxes of GPS and antenna heading angles can also serve as input boxes. If GPS or desired heading angle is known in advance, users can input it directly to the corresponding text box and pressing ”Enter”.

The map displays the geographic information of the local and remote UAVs using OpenStreetMap (OSM), an open source mapping toolkit. The red UAV icon represents the location of the local UAV, and the blue UAV icon represents the location of the remote UAV. The distance between them is displayed on the upper left corner of the map. The video display shows the real-time infrared and optical videos transmitted from the remote side. Some function keys can be used to adjust the images, such as refresh, resize and rotate. The control panel consists of six control operations, each having several widgets that facilitate user operations illustrated below:

(1) *Distance Mode Selection.* There are two modes: line-of-sight (LOS) and NLOS. In the LOS mode, the local and remote ACDA subsystems are within a short range (100-200m) and there is no obstruction between them, and hence the communication of the ACDA system can be established without alignment. In the NLOS mode, either the distance between the UAVs is beyond the short range or the signals of the two systems are blocked by obstructions in between. When LOS is chosen, the antennas alignment can be completed on the ground for optimal communication performance. When NLOS is chosen, the antenna alignment starts after UAVs are launched to the air, because attenuation, reflection, diffraction, and penetrations do not permit the alignment on the ground.

(2) *Channel Selection.* Once the "Optimize" button is clicked, the Wi-Fi channel selection algorithm described in Algorithm 1 is implemented and the optimized channels which avoid mutual interference selected on the local and remote sides are displayed in the text box to the right.

(3) *IMU Calibration.* The IMU calibration is automated. Its purpose is to eliminate magnetic interference of the environment. Once the "Run" button is clicked, the IMU calibration begins, "Run" becomes "Stop", and the text box to the right

shows "Working". One can stop the IMU calibration at any time by clicking the button again. When the IMU calibration is complete, the text box to the right shows "Done".

(4) *Initial Heading Selection.* This setting is used in a GPS denied environment such as indoors to align directional antennas by exhausting various pairs of heading angles at the local and remote sides to find the maximum RSSI. The button has two functions, "Scan" and "Stop". The text box to the right shows three states: "Default", "Working", and "Done". "Default" means that the GPS is available in current scenario, so that the initial heading selection step can be skipped. "Working" and "Done" are shown when the initial heading scan is triggered and completed, respectively.

(5) *ROS Node Operations.* Similar to the button for IMU calibration, the button of ROS nodes has two functions: "Run" and "Stop", with corresponding states, "Working" or "Done", displayed in the text box to the right. Once the ROS nodes are activated, topics are created and messages are exchanged among functional modules of the local and remote ACDA subsystems.

(6) *Cameras in Use.* Four check boxes are included for the infrared and optical cameras. Once a camera is checked, the real-time video captured by this camera will be displayed on the right. When "Save" is checked, the video will also be stored in the local laptop.

Figure 3.10 shows the flowchart of user operations.

3.2.9 Water-Resistant Enclosure Design

The water-resistant enclosure design of the ACDA system is included for the emergency response operations (see Figure 3.11). The water-resistant enclosure consists of two parts: the cover for the antenna which can rotate a full 360 degrees, and the cover for the other components, including battery, communication, computing, and control components. It supports both static seal such as static radial seal, and dynamic seal such as rotary seal. The enclosure is printed using a 3D printer with Acrylonitrile Butadiene Styrene (ABS) plastic.

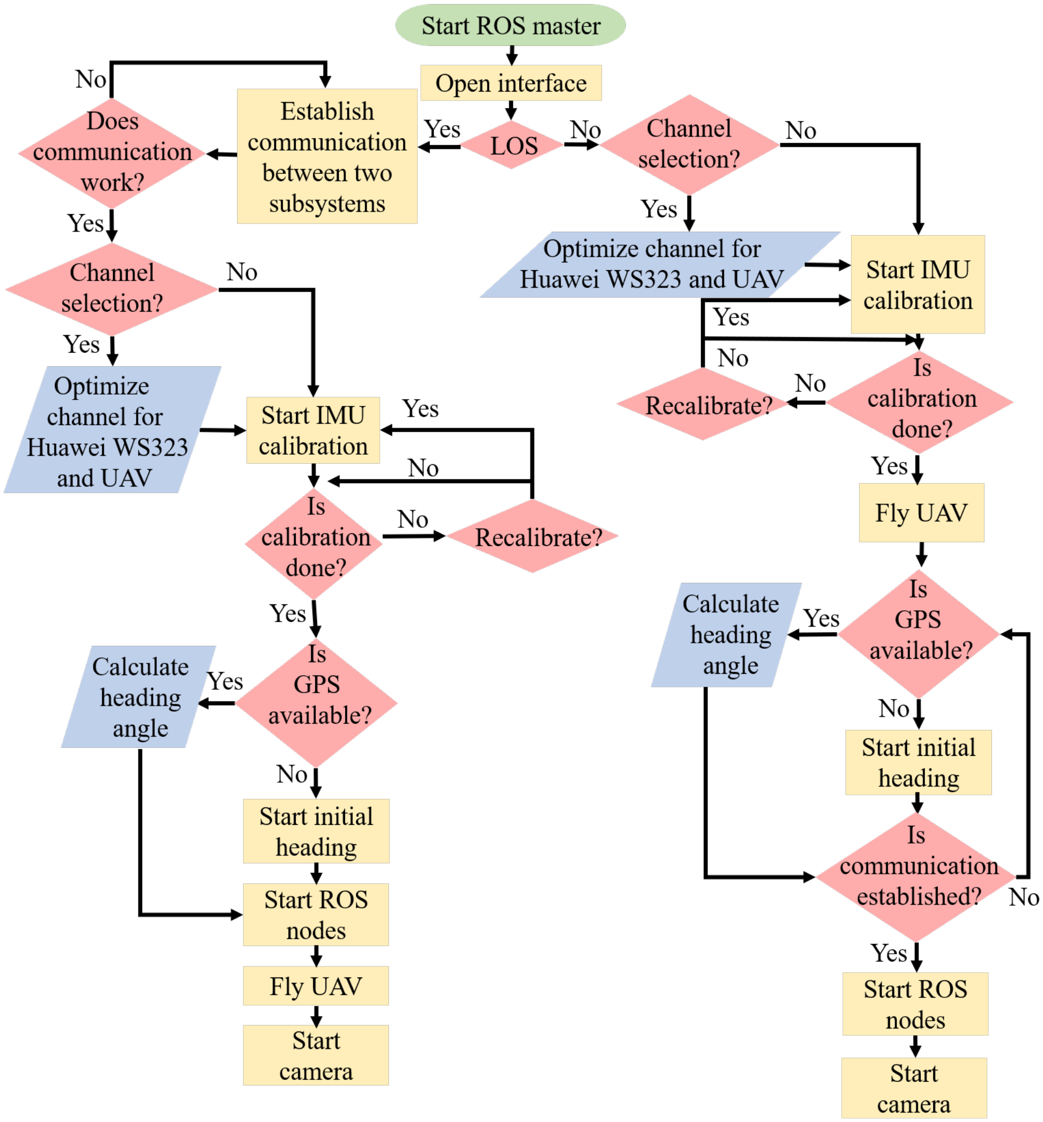


Figure 3.10. Flow chart of operations for the ACDA, coded in the user interface.

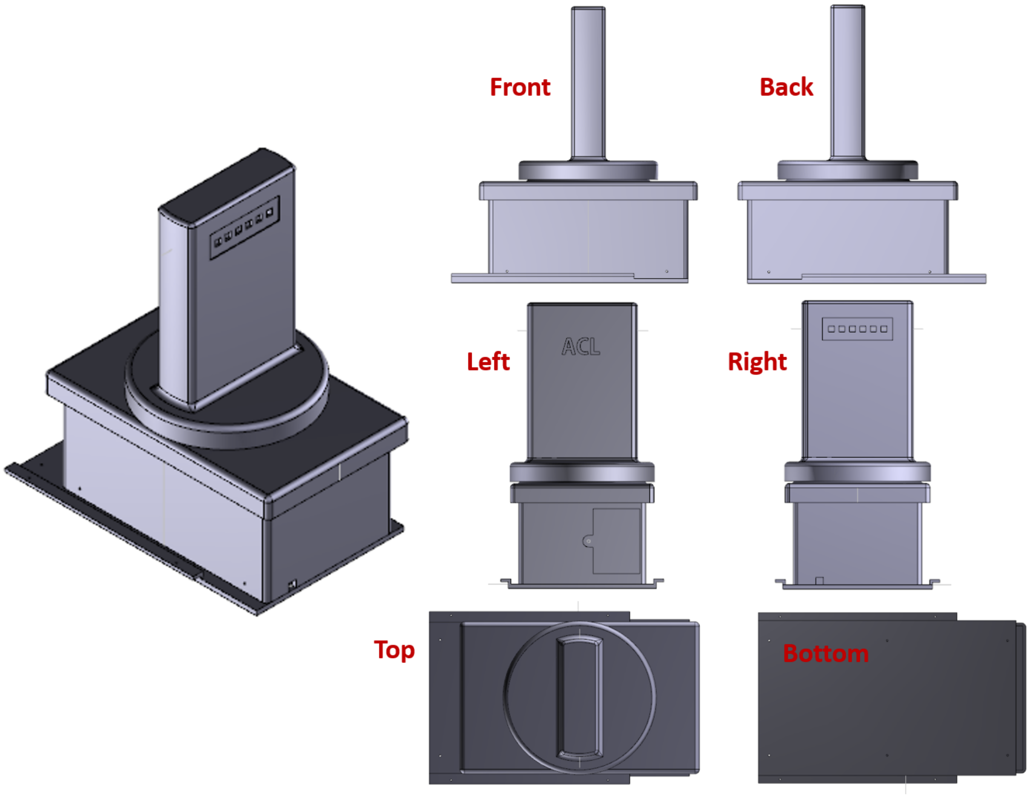


Figure 3.11. Water-resistant enclosure design.

3.3 The ACDA System Model and Controller Design

In this section, we first describe the ACDA system model including the UAV and directional antenna dynamics. We then describe the measurement models for GPS and RSSI. The antenna control problem is then formulated and an RL-based solution in an unknown communication environment is developed in the end.

3.3.1 System Models

3.3.1.1 UAV Dynamics

$$
\begin{aligned}
x_i[k+1] &= x_i[k] + v_i[k]\cos(\phi_i[k])\delta, \\
y_i[k+1] &= y_i[k] + v_i[k]\sin(\phi_i[k])\delta, \qquad (3.1) \\
\phi_i[k+1] &= \phi_i[k] + \omega_i[k]\delta,
\end{aligned}
$$

where δ is the sampling period, $\phi_i[k]$ and $\omega_i[k]$ are the heading angle and angular velocity of UAV i at time instant k respectively.

3.3.1.2 Directional Antenna Dynamics

The directional antenna installed on each UAV adjusts its heading angle automatically to establish the robust A2A communication channel. The heading angle dynamics of its directional antenna is described as

$$\theta_i[k+1] = \theta_i[k] + (\omega_i^*[k] + \omega_i[k])\delta, \qquad (3.2)$$

where θ_i is the heading angle of antennas i, and ω_i^* is the angular velocity of antennas i due to its heading control. Note that both the control of antennas i (ω_i^*) and the movement of UAV i (ω_i) contribute to the change of θ_i .

3.3.2 Measurement Models

Two measurement models are included, one for GPS, and the other for RSSI. The communication performance indicator RSSI is adopted to assist with the distributed antenna controller design.

3.3.2.1 GPS Measurement

The measurement of the GPS signal $Z_{G,i}(k)$, is described as

$$Z_{G,i}[k] = H_{G,i}(k)X_i[k] + \varpi_{G,i}[k], \tag{3.3}$$

where H_G is the measurement matrix, $H_{G,i} = [1, 0, 0, 0; 0, 1, 0, 0]$. $X_i[k] = [x_i[k], y_i[k], \phi_i[k], \theta_i[k]]^T$ is the system state of UAV i, and $\varpi_{G,i}$ is the white Gaussian noise with zero mean and covariance $R_{G,i}$. GPS signals can be transmitted through the A2A communication channel to assist with the control of the directional antennas.

3.3.2.2 RSSI Measurement

RSSI measures the performance of the communication channel [50, 51]. In the ACDA system that is equipped with two directional antennas, RSSI is affected by 1) the relative positions of the two UAVs that carry these directional antennas, 2) the headings of these antennas, and 3) the field radiation patterns of these antennas in a specific communication environment. The RSSI signal $Z_R[k]$ can be derived from the Friis free space equation as [33]:

$$\begin{aligned} Z_R[k] =& P_{t|dBm}[k] + G_{l|dBi} + 20\log_{10}(\lambda) \\ & - 20\log_{10}(4\pi) - 20log_{10}(d[k]) + \varpi_R[k], \end{aligned} \tag{3.4}$$

where $P_{t|dBm}[k]$ is the transmitted signal power, λ is the wavelength, $d[k]$ is the distance between the two UAVs at time k, and $d[k] = \sqrt{(x_1[k] - x_2[k])^2 + (y_1[k] - y_2[k])^2}$. $\varpi_R[k]$ is the white Gaussian noise with zero mean, and $G_{l|dBi}[k]$ is the sum of the transmitting and receiving antenna gains [34]. For the Ubiquiti NanoStation loco M5 directional antennas [35] that we use in the ACDA system, it is modeled based on the filed pattern of the end-fire array antennas [36]

$$
\begin{aligned}
G_{l|dBi}[k] =& (G_{t|dBi}^{max} - G_{t|dBi}^{min}) \\
& \times \sin\frac{\pi}{2n}\frac{\sin\left(\frac{n}{2}(k_a d_a(\cos(\gamma_t[k]-\theta_t[k]))-1)-\frac{\pi}{n}\right)}{\sin\left(\frac{1}{2}(k_a d_a(\cos(\gamma_t[k]-\theta_t[k]))-1)-\frac{\pi}{n}\right)} \\
& + (G_{r|dBi}^{max} - G_{r|dBi}^{min}) \\
& \times \sin\frac{\pi}{2n}\frac{\sin\left(\frac{n}{2}(k_a d_a(\cos(\gamma_r[k]-\theta_r[k]))-1)-\frac{\pi}{n}\right)}{\sin\left(\frac{1}{2}(k_a d_a(\cos(\gamma_r[k]-\theta_r[k]))-1)-\frac{\pi}{n}\right)} \\
& + G_{t|dBi}^{min} + G_{r|dBi}^{min},
\end{aligned} \tag{3.5}
$$

where $G_{t|dBi}^{max}$, $G_{t|dBi}^{min}$, and $G_{r|dBi}^{max}$, $G_{r|dBi}^{min}$ are the maximum and minimum gains of transmitting and receiving antennas. k_a is the wave number, and $k_a = \frac{2\pi}{\lambda}$. n and d_a are parameters decided by the design of the antenna. $\theta_t[k]$ and $\theta_r[k]$ are the heading angles of the transmitting and receiving antennas at time k. $\gamma_t[k]$ and $\gamma_r[k]$ are the heading angles of the transmitting and receiving antennas corresponding to the maximal $G_{l|dBi}[k]$ at time k.

The parameters $G_{t|dBi}^{max}$, $G_{t|dBi}^{min}$,$G_{r|dBi}^{max}$, and $G_{r|dBi}^{min}$ can be obtained from the antenna's data sheet in an ideal environment. In our ACDA system, $G_{t|dBi}^{max} = G_{r|dBi}^{max}$ and $G_{t|dBi}^{min} = G_{r|dBi}^{min}$ hold because the two directional antennas are of the same type. In an imperfect environment where disturbances and interference exist, these parameters in $G_{l|dBi}[k]$ can be environment-specific.

In addition, $\gamma_t[k]$ and $\gamma_r[k]$ can be obtained from the alignment of the two directional antennas [14]. In an imperfect environment, such as blockages, the desired heading angles can be captured by

$$
\gamma_r[k] = \arctan\frac{y_t[k]-y_r[k]}{x_t[k]-x_r[k]} + \theta_{env}, \tag{3.6}
$$

where $(x_t[k], y_t[k])$ and $(x_r[k], y_r[k])$ are the positions of UAVs that carry the transmitting and receiving antennas respectively, and θ_{env} is an environment-specific shift angle. $\theta_{env} = 0$ in a perfect environment.

3.3.3 Problem Formulation

Our goal is to find the directional antennas' optimal heading angle velocities to maximize the RSSI performance over a look-ahead window, and thus, to maximize the performance of the UAV-to-UAV communication channel. The RSSI model (as described in Equations (3.4)-(3.6)) contains unknown environment-specific parameters, i.e., $G^{max}_{t|dBi}$, $G^{min}_{t|dBi}$, and θ_{env}.

Here we formulate the problem as an optimal control problem. Mathematically, considering the system dynamics described in Equations (3.1) and (3.2), an optimal control policy $U[k]$ is sought to maximize the following value function

$$V(X[k]) = \sum_{l=k}^{k+N} \alpha^{l-k} Z_R[l](X[l], U[k]), \tag{3.7}$$

where $X[k]$ is the global state, $X[k] = [X_1^T[k], X_2^T[k]]^T$, $X_i[k] = [x_i[k], y_i[k], \phi_i[k], \theta_i[k]]^T$. $U[k]$ is the control input, $U[k] = [U_1[k], U_2[k]]^T$, $U_i[k] = [\omega_i^*[k]]$. $\alpha \in (0, 1]$ is a discounter factor, and $Z_R[l]$ is the RSSI signal at time l. Note that the control is decentralized. Each directional antenna finds its own optimal control policy with the assumption that the other antenna adopts its optimal control policy.

Next we develop the control solution for one of the UAVs (denoted as the local UAV, or UAV 1). The control solution for the other UAV (the remote UAV, or UAV 2) is designed in the same manner.

3.3.4 Reinforcement Learning Based Optimal Control for ACDA

To solve the optimal control problem formulated above, an on-line adaptive optimal controller is developed based on reinforcement learning.

Note that the value function (described as Equation (3.7)) can be rewritten as

$$\begin{aligned} V(X[k]) =&Z_R[k](X[k],U[k]) \\ &+\sum_{l=k+1}^{k+N} \alpha^{l-k} Z_R[l](X[l],U[k]) \end{aligned} \tag{3.8}$$

To solve the above Equation online, here we utilize the RL method, in particular, the policy iteration (PI) algorithm [27, 28]. The PI algorithm contains two steps: policy evaluation and policy improvement. The policy evaluation step is designed to solve the value function $V(X[k])$ using Equation (3.8), given the current control policy. The policy improvement step is to maximize the value function by finding the optimal control policy. The two steps are conducted iteratively until convergence.

Policy Evaluation

$$\begin{aligned} V_{j+1}(X[k]) =&Z_R[k](X[k],U[k]) \\ &+\sum_{l=k+1}^{k+N} \alpha^{l-k} Z_{j,R}[l](X[l],U[k]) \end{aligned} \tag{3.9}$$

Policy Improvement

$$\begin{aligned} U_{j+1}(X[k]) =&\arg\max_{U_j[k]} Z_R[k](X[k],U[k]) \\ &+\sum_{l=k+1}^{k+N} \alpha^{l-k} Z_{j+1,R}[l](X[l],U[k]) \end{aligned} \tag{3.10}$$

where j is the iteration step index, and $Z_{j,R}[l](X[l],U[k])$ is the RSSI model with parameters learned in the jth iteration step.

Note that three unknown parameters for the environment-specific RSSI model ($G_{t|dBi}^{max}$, $G_{t|dBi}^{min}$ and θ_{env}) are involved in Equation (3.9), and need to be learned. In particular, for each iteration j, three time steps (k, $k+1$ and $k+2$) are needed to come up with three equations to iteratively solve for the three parameters. To solve the nonlinear equations, we utilize the Newton's method [57]. The idea of the Newton's method is described as follows. An initial guess which is reasonably close to the true root is first given, and the function is then approximated at the given initial guess by its tangent line. After computing the x-intercept of this tangent line, which is typically a better approximation to the function's root than the original guess, the function is approximated again at the derived x-intercept until the accuracy meets the requirement.

3.4 Experimental Results

3.4.1 Simulation Studies

In this section, we conduct simulation studies to illustrate and validate the antenna controller design. Two UAVs are simulated to move in a 2-D airspace, each equipped with a directional antenna. The total simulation time is $T = 45s$, with the sampling period $\delta = 1s$. The parameter N in Equation (3.7) is selected as 1 here (i.e., with the goal of maximizing the current RSSI). The transmitting power is $P_{t|dBm} = 23dBm$, and the electromagnetic wavelength is $\lambda = 0.052m$. For the parameters in Equation (3.5), we select the design parameters $n = 8$ and $k_a = \frac{\lambda}{4}$. Figures 3.12 and 3.13 show the trajectories of local UAV and remote UAV respectively. The blue solid curve and the red dotted curve are the real trajectory and the GPS measurements respectively. Gaussian noises are added to the GPS signals.

With these randomly generated UAV trajectories, we simulate the RL-based control algorithm. To simulate the long-distance communication scenario, the minimum received signal strength is assumed to be 0, and in this case, the minimum directional antennas' gain ($G_{t|dBi}^{min}$) can be calculated accordingly. Figures 3.14 and 3.15 show the learned environment-specific antennas' maximum gain ($G_{t|dBi}^{max}$) and the shift angle caused by the environment (θ_{env}) respectively. Gaussian noises are added to the RSSI measurements. As shown in the figures, the learned parameters are very close to their true values, which indicates the effectiveness of the learning algorithm. Figures 3.16 and 3.17 show the derived optimal heading angles of the local directional antenna and the angle errors between the derived optimal heading angle and the true optimal heading angles. The derived optimal heading angles are very close to the true optimal angles, indicating the good performance of the RL-based control algorithm.

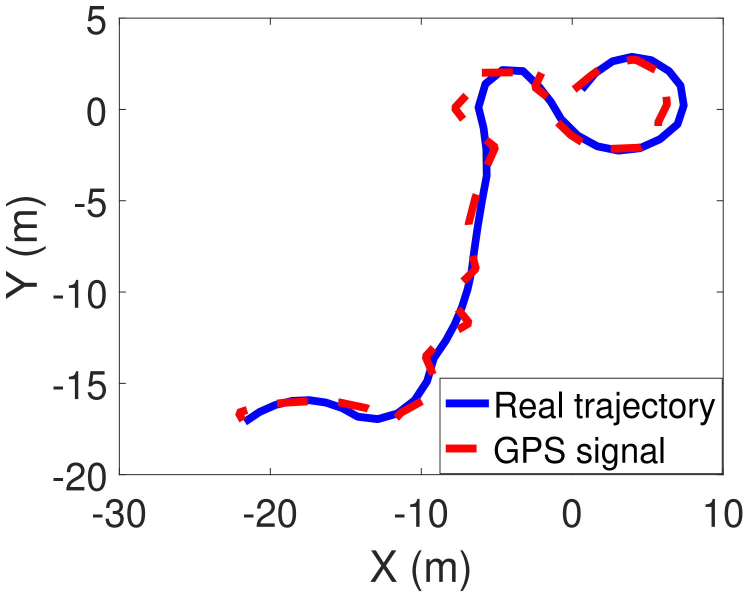


Figure 3.12. Trajectory of UAV 1.

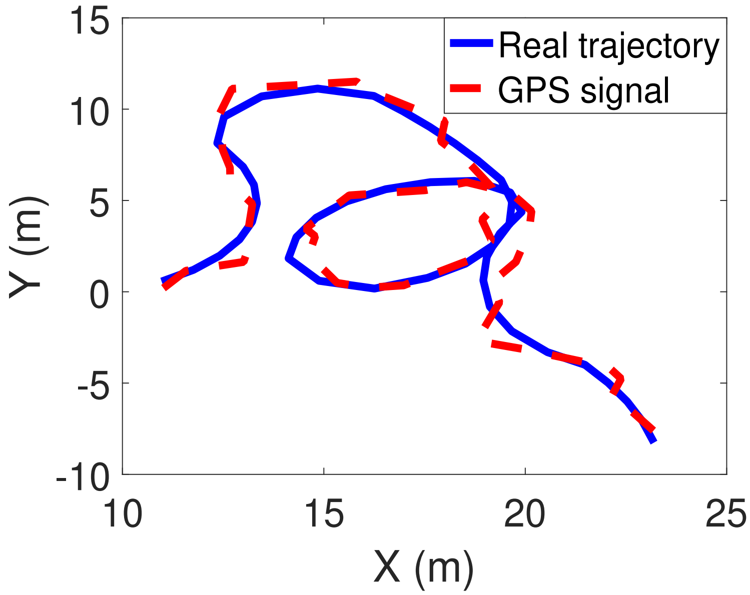


Figure 3.13. Trajectory of UAV 2.

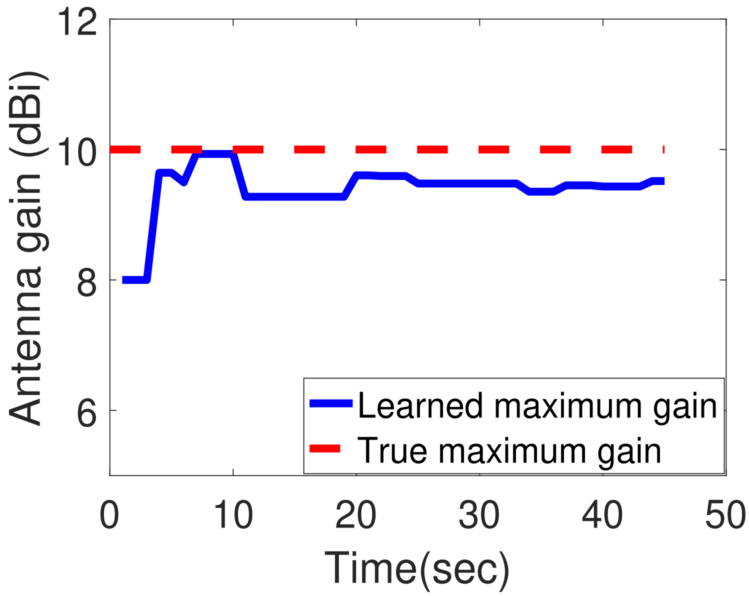


Figure 3.14. Learned environment-related — maximum directional antenna gain.

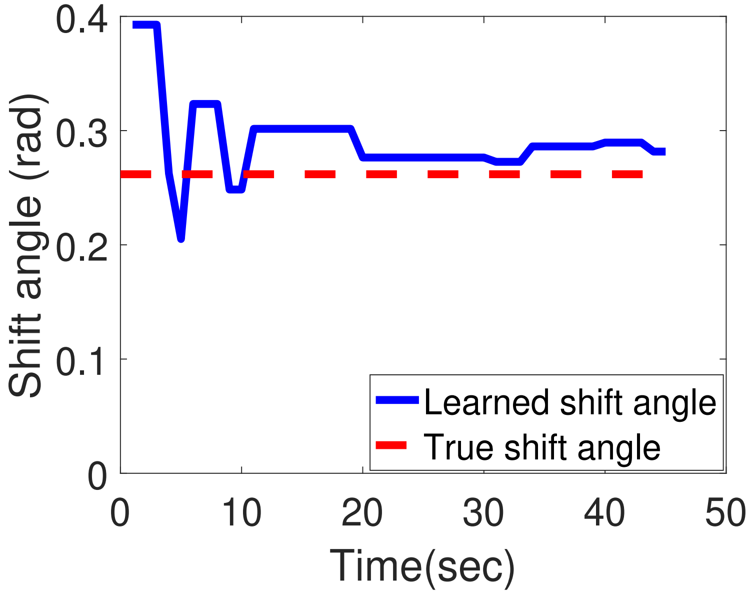


Figure 3.15. Learned environment-related — shift angle.

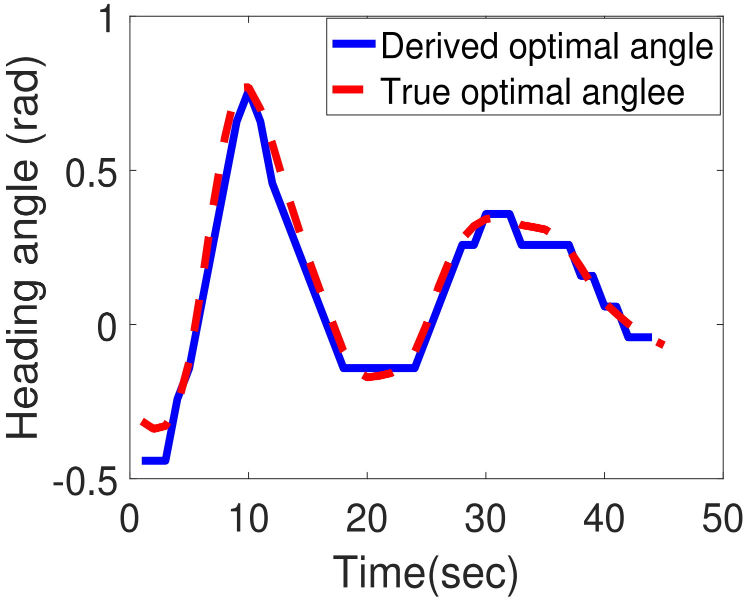


Figure 3.16. Derived optimal heading angles.

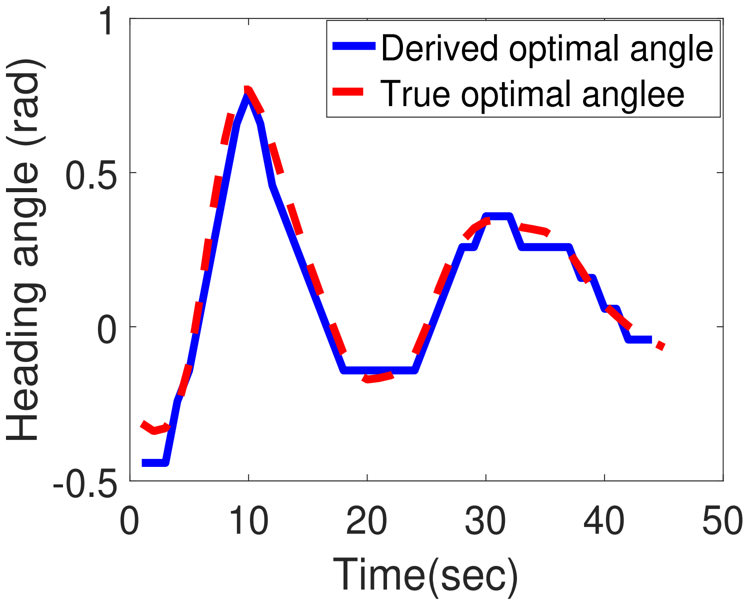


Figure 3.17. Heading angle errors between the derived angles and the true optimal angles.

3.4.2 Field Tests

We conducted two field tests to verify the proposed ACDA solution. The first is to verify the RSSI model, and the second is to test the performance of the antenna control algorithm through a comparative study with the sole GPS-based control algorithm described in [13].

3.4.2.1 Testing of the RSSI Model

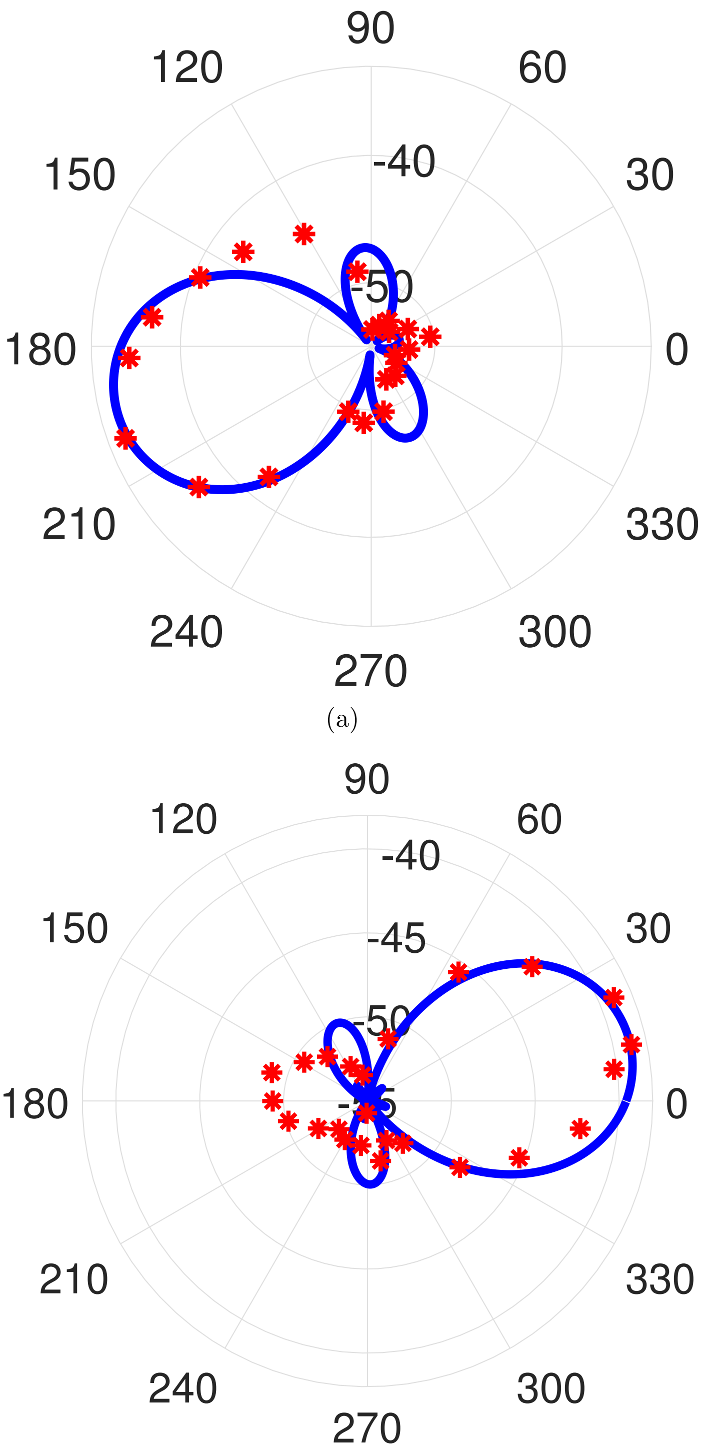
We first test the RSSI model for a pair of directional antennas, as described in Equations (3.4) and (3.5). The distance between the two directional antennas is $43m$, the transmitting power is $23dBm$, and the electromagnetic wave frequency is $5.8GHz$. With the same environment and test settings, we conduct two tests to check the properties of the two directional antennas. In the first test, the remote antenna points towards the local antenna and keeps a fixed heading angle throughout the test.

The local antenna initially points towards the remote antenna and then rotates 15° per $90s$ until 360°. The RSSI is collected by the local antenna and is averaged during each $90s$. In the second test, the heading angle of the local antenna is fixed, and points towards the remote antenna. The remote antenna initially points towards the local antenna and rotate 15 degrees per 90 seconds until 360°. The RSSI collected by the remote antenna is also averaged in the second test. Parameters in the RSSI model are estimated from the measured data: $n = 8$, $d_a = \frac{\lambda}{4}$, $G_1^{max} = 11.5dBi$, $G_1^{min} = -8.5dBi$ for the local antenna, and $G_2^{max} = 9dBi$, $G_2^{min} = -8dBi$ for the remote antenna. Figures 3.18(a) and 3.18(b) show the test results. The tested RSSI mappings match reasonably with the proposed model.

3.4.2.2 Testing of the Control Algorithm

We also test the proposed RL-based control algorithm. In this preliminary testing, the local directional antenna is at a fixed location, and the remote antenna changes its location every $90s$. Each antenna finds its own optimal heading angle at each location according to the distributed RL-based control algorithm described in Chapter 3.3 to optimize the RSSI performance. The RSSI is measured at both antennas with the derived heading angles. To provide a comparison, we also test the GPS alignment-based control algorithm proposed in [13], in which each antenna points toward the GPS location of the other controller, and RSSI is not used as a measurement signal. The testing results are summarized in Table 1. At each location, two control algorithms are tested. ”RL” in Table 1 represents the RL-based control algorithm, and ”GPS” means the GPS alignment-based control algorithm.

The proposed RL-based algorithm demonstrates a better RSSI, as compared to the GPS alignment-based algorithm, in all five locations. The testing results verifies



(a)

(b)

Figure 3.18. RSSI mapping in (a) the first test, and (b) the second test. The red dots are the measured RSSI signal, and the blue curves pictures the relation between the RSSI signal and the rotation angle according to Equations (3.4) and (3.5).

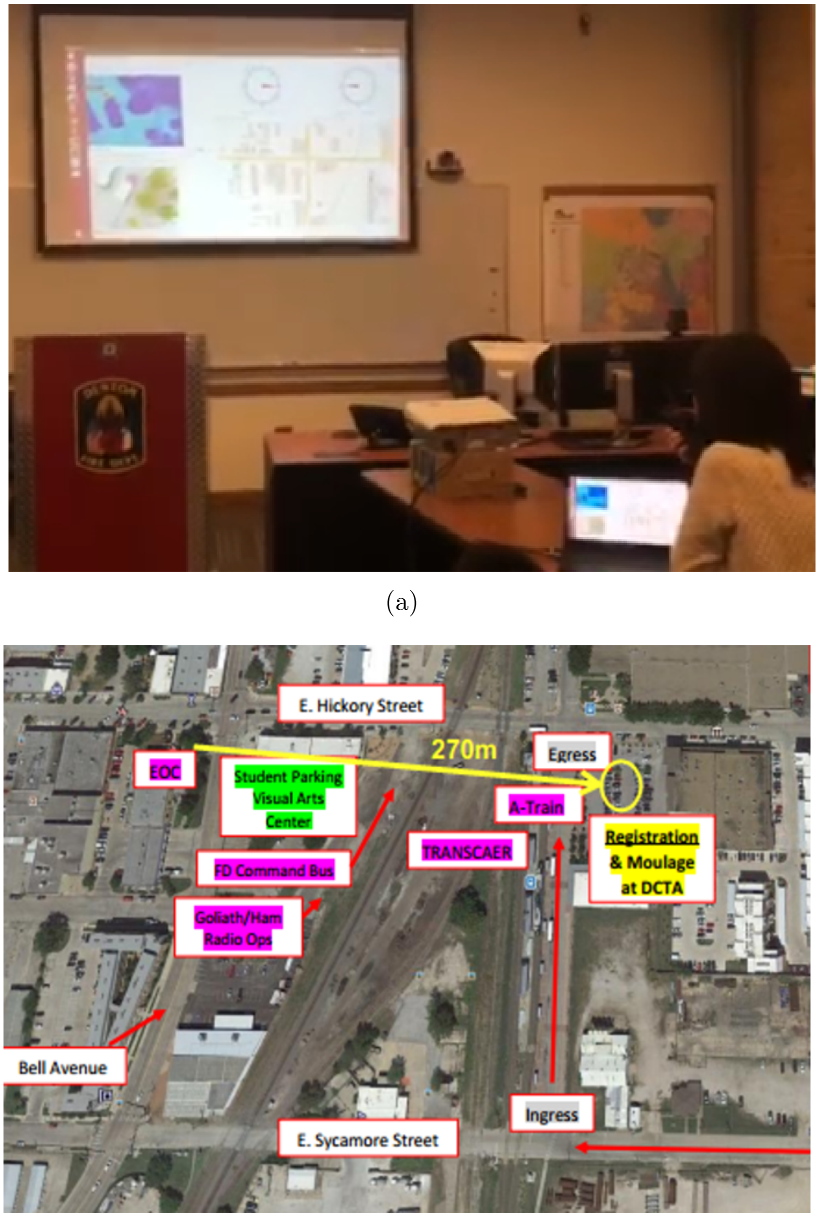
Table 3.1. Comparative Field Testing Results

		Local antenna		Remote antenna	
Position	Control	RSSI	Heading	RSSI	Heading
1	RL	-37dBm	194.1°	-41dBm	16.4°
	GPS	-45dBm	170.6°	-45dBm	35.5°
2	RL	-37dBm	197.3°	-39dBm	15.1°
	GPS	-39dBm	176.2°	-42dBm	357.8°
3	RL	-39dBm	191.1°	-44dBm	13.2°
	GPS	-41dBm	182.6°	-44dBm	6.2°
4	RL	-35dBm	196°	-39dBm	15.2°
	GPS	-38dBm	195.8°	-39dBm	16.2°
5	RL	-35dBm	194.9°	-39dBm	15.4°
	GPS	-37dBm	186.1°	-39dBm	7.5°

that the proposed RL-based control algorithm outperforms the GPS alignment-based algorithm.

3.4.2.3 Emergency Drill

We tested the performance of our ACDA system through participating in a full-scale emergency drill of the City of Denton, TX. On a raining day in May 2018, a full-scale disaster drill on a train-involved accident was conducted with the lead of the Denton Fire Department. An incident occurred on the Union Pacific rail line and passengers on the Denton county transportation authority (DCTA) A-Train piled out of the train with a great panic, and caused injuries and fatalities. Our ACDA system successfully responded to the emergency and sent back real-time monitoring videos of the emergency zone to the emergency management center for condition accessment (see Figure 3.19).


(a)

(b)

Figure 3.19. Emergency drill with Denton Fire Department. (a) The room of the emergency management center and the screen where the remote infrared and optical monitoring videos are transmitted and displayed. (b) The map of this disaster drill. The distance between the local and the remote UAVs is about 270m, beyond the nominal Wi-Fi range.

CHAPTER 4

Beyond Visual Line of Sight UAV Control for Remote Monitoring using Autonomously Aligned Directional Antennas

4.1 Introduction

UAVs have found broad commercial use in civilian applications. For instance, UAV-assisted nondestructive health monitoring has been used to monitor bridge infrastructures to avoid catastrophic accidents [58–61]. Traditional infrastructure inspections are often performed through a manual visual-based procedure, which are expensive, time-consuming, and even unsafe for some structural parts, e.g., the sides and underneath parts of bridges. Paper [58] studied the feasibility of using UAVs for fatigue crack detection in bridges and demonstrated the field performance using three different UAV types. In [59], the authors analyzed different UAV types based on their weights, flight type, payload and flight time, and selected a rotary-wing typed drone for crack detection of bridges. Paper [60] used an on-board high definition (HD) camera to collect images of bridges and stored the images in a storage card for further processing and crack detection. Paper [61] used a UAV for concrete crack identification. The UAV communicated images, distance information and control signals with a human operator through a Wi-Fi module.

In all existing UAV applications, UAVs are controlled within the visual line of sight. This restriction limits the flexibility and applicability of UAVs. Many UAV applications require the beyond BVLOS control. For instance, in emergency response, UAV operators often cannot get close to emergency zones because of broken roads, debris, or remaining dangers in the zones [8, 9]. The standard communication range, such as Wi-Fi, is not sufficiently long for UAV control. Similarly, in nondestructive health monitoring, an operator may not always be able to keep the UAV within visual sight due to various structural constraints [62]. Blockages should not limit UAV operations.

To address the aforementioned issues, we here aim to develop a BVLOS control solution for UAVs, which extends the operation range to kilometers, and is not limited by blockages between an operator and a UAV. Designing BVLOS UAV control is challenging, considering difficulties such as UAV situation awareness and wireless communication robustness. The research on BVLOS control is limited [63]. Paper [63] describes a transportation system for UAV BVLOS applications based on LTE. The solution is expensive and requires the existence of LTE support. There is a need to implement UAV BVLOS control using a communication system that is cost-effective and can be quickly deployed without ground infrastructure support.

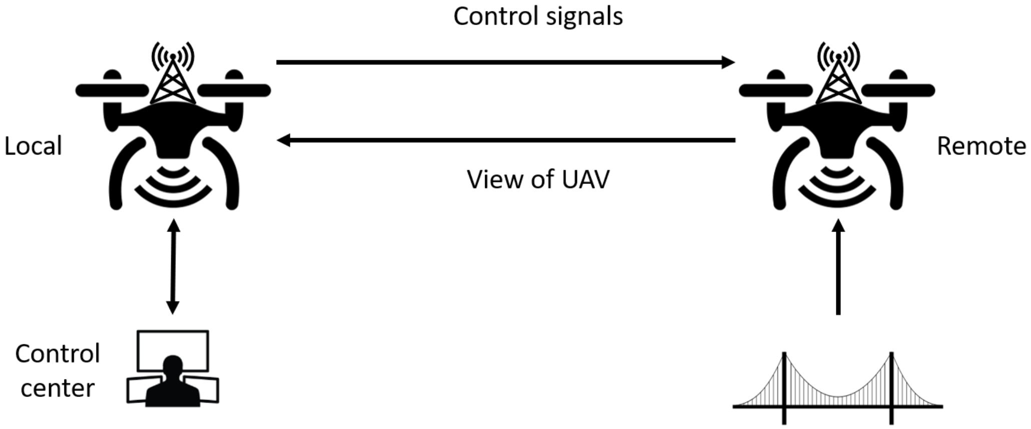


Figure 4.1. Illustration of the teleoperation of the remote UAV based on the ACDA system .

The BVLOS UAV control solution we design in this chapter is shown in Figure 4.1. The solution is based on the long-distance and broadband ACDA system that we developed previously [11–13, 20, 21, 40, 41]. In [13], we designed and implemented the complete ACDA system, which is composed of two UAV platforms (one for the local side, and the other for the remote side), on-board autopilot, communication component, and autonomous antenna heading alignment. The local UAV platform serves as the network relay and the remote UAV platform is used for monitoring. The two UAVs are dispatched to the air to establish communication, and then the local UAV communicates with the local ground device through an UAV-ground communication channel. The ACDA system does not rely on ground infrastructure support, and hence can be applied to scenarios where fixed communications infrastructures are not available. With the use of directional antennas and an automatic alignment algorithm, the ACDA system features long-range communication and board throughput. In [21], we improved the ACDA system with a unified communication and control channel, integrated design and implementation of communication, control and computing components, and a user-friendly interface. Moreover, we designed a reinforcement learning-based directional antenna control algorithm that maximizes communication performance in unknown communication environments.

The aforementioned works still do not realize BVLOS UAV control for the following reasons. First, in the ACDA system, each UAV is controlled locally by a UAV operator. UAV control is within line of sight. In this chapter, we transmit the remote UAV's control signal through the ACDA system to enable BVLOS operations. Second, according to our experimental studies, the ACDA system has some performance limitations in terms of interference and endurance. In this chapter, we redesign the ACDA system to overcome these performance limitations. Our contributions are summarized as follows.

The first contribution is the design of BVLOS UAV control, using the ACDA system with a local UAV relay. The use of UAV relay permits the transmission of flight control signals even if blockages exist between the UAV and the operator. The use of directional antennas extends the UAV control distance to kilometers with high data throughput. The BVLOS control solution includes a two-way relay communication. In one direction, low-resolution videos from the remote UAV is transmitted through the local UAV relay to help UAV the operator navigate the remote UAV. The low resolution videos are for navigation. They provide situation awareness to UAV operators and allow operator to locate targets of interest, such as structural defects in health monitoring and victims in emergency response. In the other direction, UAV control signals are relayed through the local UAV and transmitted to the remote UAV for BVLOS navigation. High-resolution videos from the remote UAV can be stored in a storage card for further processing. We design a remote control solution to operate the remote UAV using a keyboard at the local side of the ACDA system.

The second contribution lies in the ACDA redesign to enhance its performance. In particular, the microprocessor is changed form BeagleBone Black [52] to NAVDIA Jetson TX2 Module [64] (TX2) which has a better computing capability for BLOVS operation. In addition, the UAV platform is changed from Matrix 100 to Tarot 650 [65] with the open source flight controller Pixhawk [66] to remove the interference between the UAV platform (and in particular carbon fiber propellers) and the communication system. The redesign also includes the rotational structure of directional antennas to improve its endurance with 360-degree rotation capability. Finally, the propellers and the motors are reconfigured to enable 25 minutes of flight time.

The rest of this chapter is organized as follows. Chapter 4.2 describes the upgraded ACDA system design in terms of both hardware and software design. Chapter 4.3 describes the BVLOS UAV control using a keyboard through the ACDA relay and includes the verification studies using simulation studies and field tests.

4.2 ACDA System Design

This section introduces the ACDA system and describes the design of the upgraded ACDA system in detail.

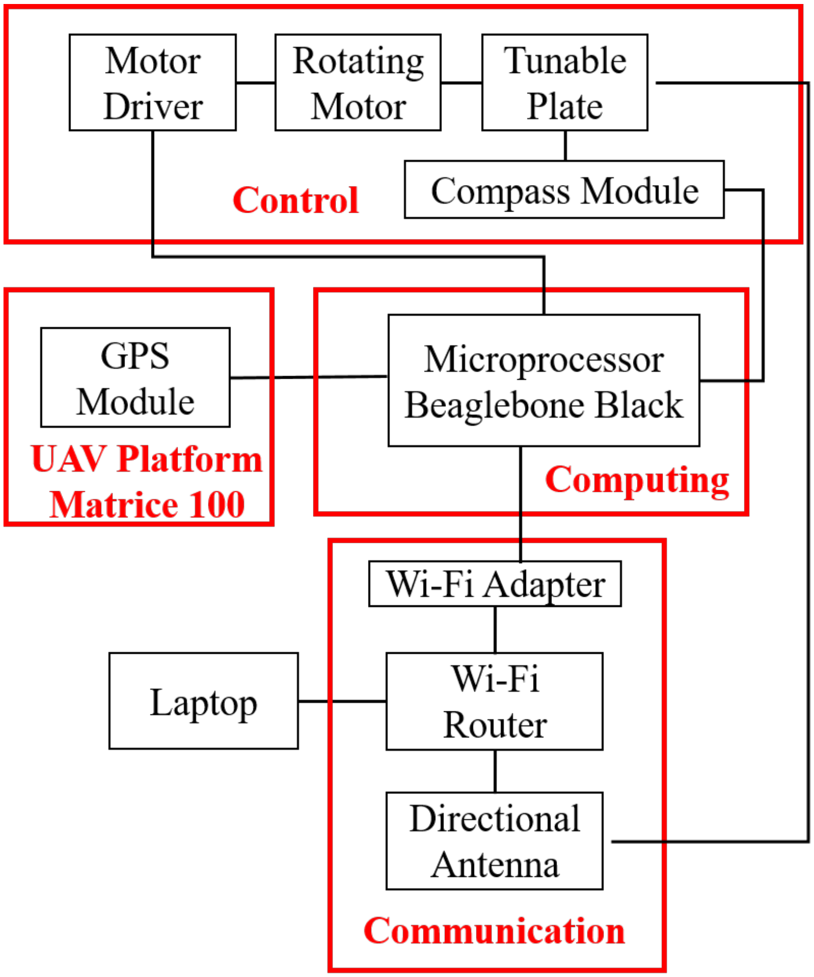


Figure 4.2. The Previous ACDA system.

4.2.1 Overview of the ACDA System

In [21], we designed and implemented an ACDA system. The ACDA system is composed of a pair of subsystems on both the local and the remote side. The

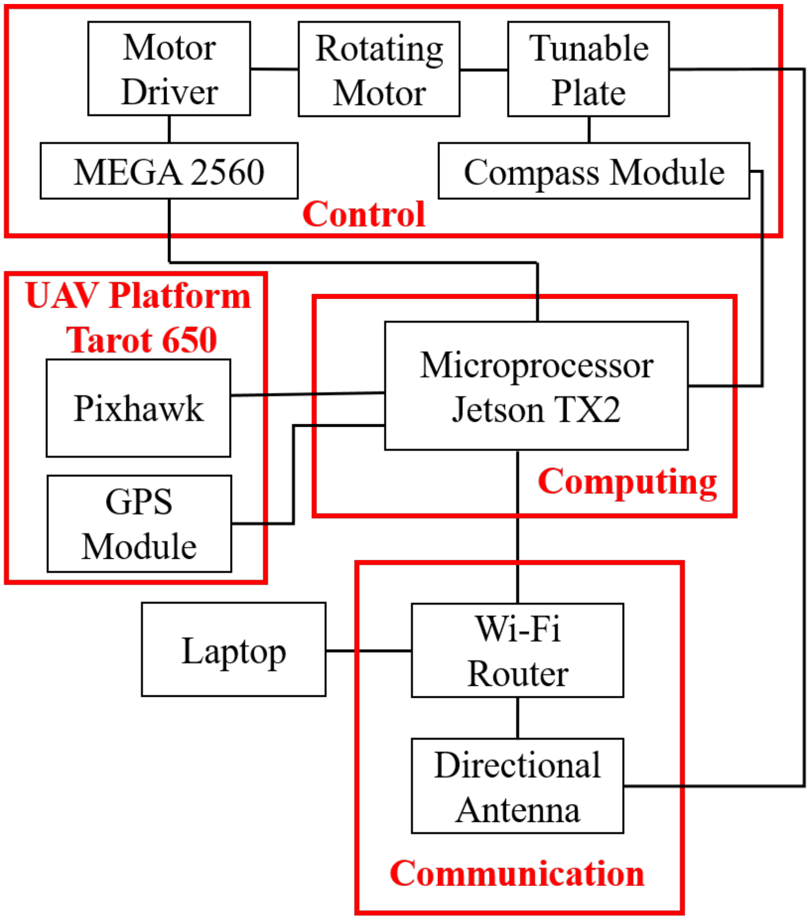


Figure 4.3. The new ACDA system.

subsystem on each side is composed of four parts: UAV platform, control, computing component, and communication component (shown in Fig 4.2). The communication component is critical for the ACDA system. It includes a Wi-Fi adapter, a Wi-Fi router and a directional antenna as shown in Fig 4.4. The types of links include: A2G/G2A, A2A, and A2C/C2A. The control component rotates the directional antennas automatically using an alignment algorithm so that the system can achieve its best communication capacity. The computing component is responsible for collecting data, processing the computing and learning tasks, and exporting motor control signal to the control component. With an user-friendly interface installed on the local laptop, the ACDA system configuration and system information (the locations of UAVs, distance, communication quality, antenna alignment performance and real-time videos) can be displayed distinctly.

This chapter describes the upgrade to the ACDA system for BVLOS operation (see Figure 4.3). To increase the stability and improve communication performance, we change the UAV platform to Tarot 650 with Pixhawk. To improve the computing capability, we use a TX2 for the computing component instead of the BeagleBone Black. To achieve better control performance, we adopt a lazy susan bearing component and an ATmega2560 chip for the control component. In addition, a new battery is assembled to expand the working time of the ACDA system. The following subsections discuss each improvement in detail.

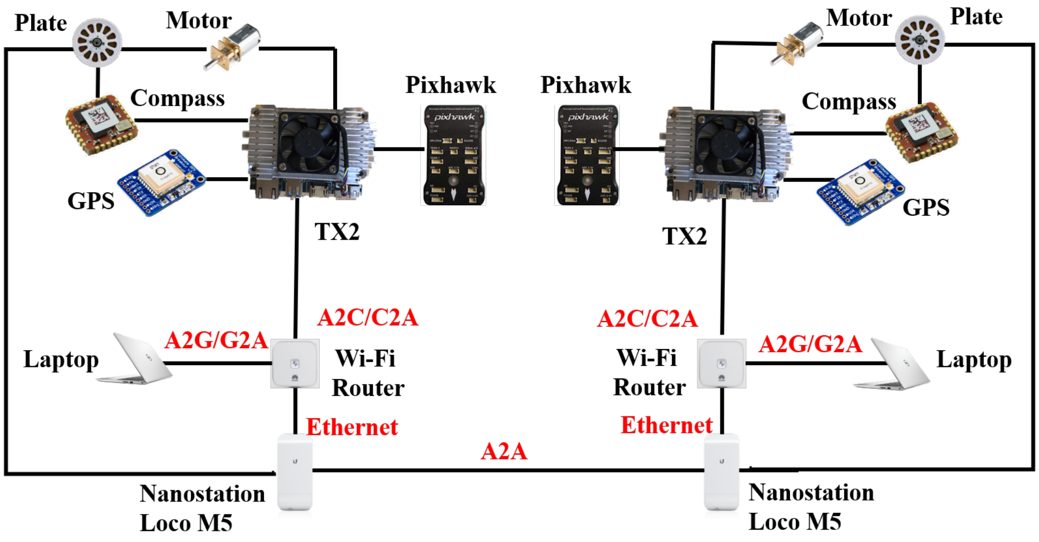


Figure 4.4. Illustration of the connections among different components in the ACDA system.

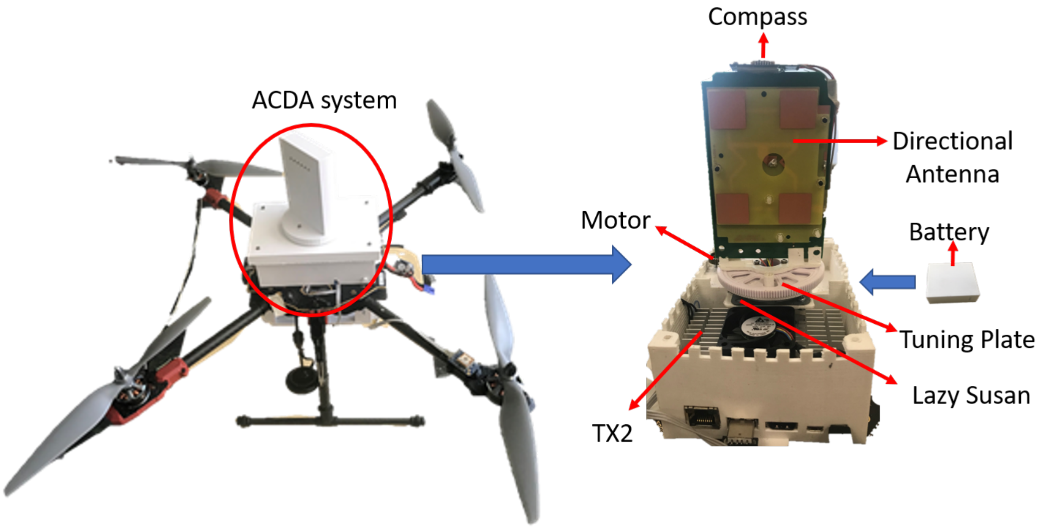


Figure 4.5. Tarot 650 with ACDA system.

4.2.2 Upgraded ACDA System Design

4.2.2.1 The UAV Platform

Considering the performance of UAVs such as payload, flight time, expandability, stability, and operability, Matrice 100 [21] and Tarot 650 with Pixhawk [13] (see Figure 4.5) are two good options. The flight control signal at $2.4GHz$ and carbon fiber propellers of Matrice 100 can potentially interfere with the WiFi router, which also operates at $2.4GHz$, and this interference can cut in half the throughput of the communication capacity. To avoid this interference, we choose Tarot 650 with plastic propellers ($13inch$) as the UAV platform for the new ACDA system. The flight control signal link of Pixhawk operates at $72MHz$, which does not overlap with the Wi-Fi router. The battery for Tarot 650 is a LiPo battery of $10000mAh$. In addition, the Pixhawk's TELEM2 (a serial port) is connected with a universal serial bus (USB) port of TX2 for communication between these two devices.

4.2.2.2 Communication System

The TX2 module has a built-in WiFi unit, and hence no external WiFi adapter is needed. The ACDA system has A2G/G2A, A2A, A2C/C2A communication links as shown in Figure 4.4. Two directional antennas form an A2A link, with an A2G/G2A link on each side using a WiFi router. The WiFi router also forms an A2C/C2A link with the WiFi module in TX2. Using these three types of wireless communication links, TX2s collect data from sensors and Pixhawks and publish the data to the ACDA system using the ROS.

4.2.2.3 Computing Component

The function of the computing component is to obtain data from the sensors and Pixhawk, publish obtained data to ROS, calculate the desired heading of antennas and send the motor's rotation direction and speed to the control component of the ACDA system. The computing component is implemented using TX2. We use TX2 instead of the Beaglebone Black used in [21] because of its better computing capability. The TX2's CPU includes a dual-core NVIDIA Denver2 ($2GHz$) and a quad-core ARM Cortex-A57 ($2GHz$). Its GPU is 256-core Pascal. Its CSI2 supports $2.5Gbps/Lane$. Meanwhile, it has $8GB$ $128bit$ LPDDR4 memory and $32GB$ eMMC onboard flash storage. Because the original NVIDIA carrier board of TX2 is too heavy for the UAV platform, we design a customized carrier board for TX2 with a size of $88mm \times 65mm$ and a weight of $53g$ (see Figure 4.6).

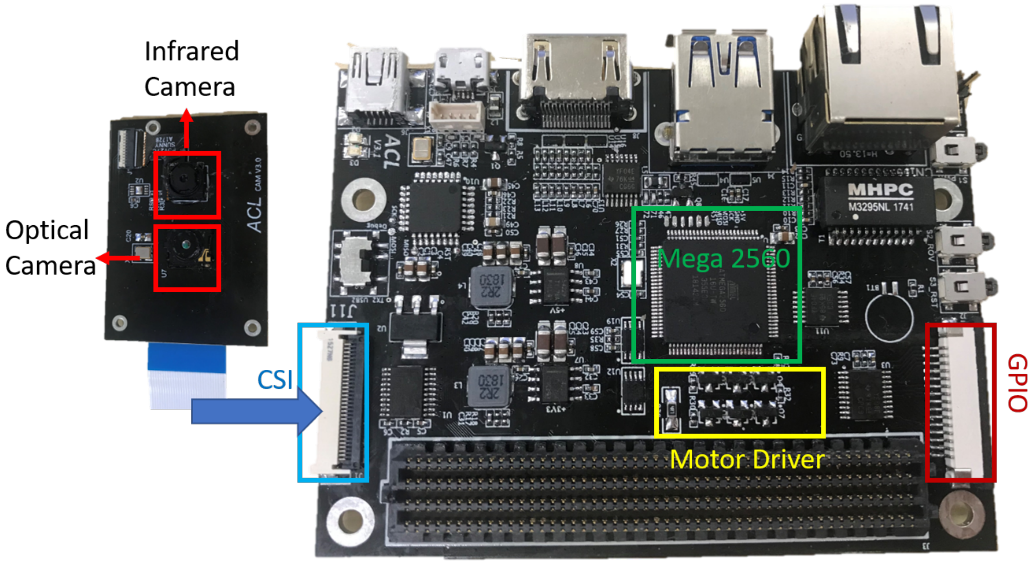


Figure 4.6. TX2 customized carried board.

4.2.2.4 Antenna Heading Control Component

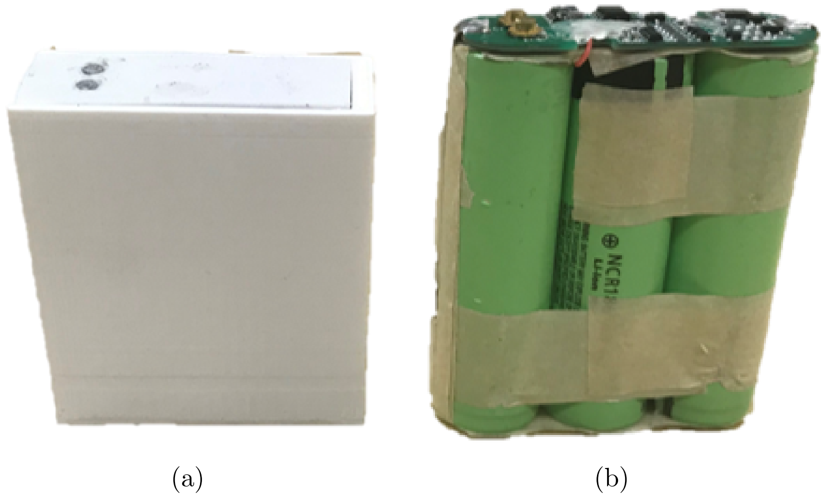
The control components rotate the geared motors to align the two directional antennas. The control component consists of a ATmega2560 chip [67], an H-bridge motor driver, a geared DC motor, a magnetometer and a rotatable plate with a lazy susan turntable (see Figure 4.5). The ATmega2560 reads the location from the local GPS module and RSSI data from a directional antenna and then sends data to the local TX2. The ATmega2560 receives motor control signals from TX2 and outputs stable PWM signals to the motor driver to rotate the motor. The ATmega2560 and the motor driver are integrated on a TX2 carrier board to save space and reduce wires between components (see Figure 4.6). Furthermore, ATmega2560 can output stable PWM signals whereas TX2 cannot output and expand I2C, UART, and SPI ports for TX2. TX2 connects ATmega2560 via USB ports. We select a lazy susan turntable to connect the rotational parts to non-rotational parts of the system and use a slip ring to connect TX2 with a directional antenna and a compass. Compared with the previous rotational design in [21], this new design guarantees the coaxiality of the tuning plate to avoid shaking. The design has a more stable and smoother antenna rotation than the previous design.

4.2.2.5 Autonomous Antenna Alignment

Because of the mobility of UAVs, it is necessary to automatically align two antennas to maintain the robust communication and achieve the best communication capacity. This is achieved through an antenna heading control algorithm. Antenna heading control aims to autonomously diminish the difference between the desired and currently the measured headings. The desired heading is calculated from local GPS and remote GPS locations of both UAVs for antenna alignment. The current heading is a direct output of compass. The difference is sent to the heading control algorithm [13], which is a linear quadratic gaussian controller consisting of a Kalman filter and a linear quadratic regulator. The Kalman filter is used to estimate the system states $\hat{x}$. The lease-quadratic regulator is a state feedback controller. The LQG is to minimize the quadratic GPS-based tracking error. The LQG controller can be described as $p^* = -K_c\hat{x}$, where K_c is the gain matrix of the optimal controller and is sent to the control part to rotate the directional antenna assembly.

4.2.2.6 Battery Design

The space and payload of the UAV platform restrict the components' size and weight. Therefore, a small-size, lightweight and high-capacity battery is desired for the ACDA system. Because We can not find batteries in the marketplace with the shape and size suitable for the ACDA system. We design a new rechargeable battery using three Panasonic Li-ion MH12210 and a charge protection circuit board (see Figure 4.7). The output voltage is $12.6V$ and the total capacity is $3250mAh$. The



(a) (b)

Figure 4.7. (a) battery box. and (b) three cascaded batteries with a rechargeable protective circuit.

battery can support the ACDA system to work for up to 2 hours, and enhances the durability of the system.

4.2.2.7 Information Flow in ROS

ROS is used to transmit messages between multiple devices in the ACDA system (see Figure 4.8). TX2 communicates with MEGA2560 using the Rosserial package [68] and with Pixhawk using the MAVROS package [69]. TX2 directly reads the directional antenna heading from a compass module. TX2 subscribes GPS and RSSI ROS topics from ATmega2560 and publishes motor PMW control signals to ATmega2560. In addition, one optical camera (OmniVision OV5693) and one infrared camera (FLIR LEPTON 3) are connected to the remote TX2 via a camera serial interface (CSI). The remote TX2 first launches a ROS video node to obtain a high resolution (up to 2592×1944) image from OV5693. Furthermore it saves the high resolution image at the remote TX2. Meanwhile, it converts the high resolution image to a low resolution image (640 × 480) which is subscribed and displayed at the interface on a local laptop. The low resolution images are used for BVLOS UAV control.

For BVLOS UAV control, MAVROS must be executed in the remote TX2 to convert MAVLink messages of Pixhawk to ROS messages. In addition, Pixhawk also subscribes control signals from the remote TX2. The remote TX2 collects videos from a optical camera and an infrared camera and converts videos to ROS image topics. All ROS topics are published in a ROS network. The local Laptop subscribes all the messages and displays them on the user-friendly interface that was introduced in [21]. The local laptop also publishes the control ROS topic to the remote UAV.

4.3 BVLOS UAV Control

This section describes the development of a BVLOS solution to control the remote UAV from the local side using a keyboard.

4.3.1 System Design and Implementation

4.3.1.1 Pixhawk Setup

To enable communication between TX2 and Pixhawk, MAVLink, a message protocol, needs to be enabled on a configurable serial port of Pixhawk. The TELEM 2 (serial port) port is selected to link with a USB port of TX2. Nevertheless, the default setting of TELEM 2 port is disabled in Pixhawk's open source flight control software PX4 (V1.9.0). Therefore, we first need to enable the TELEM 2 port using PX4 ground station QGoundControl (QGC), which provides full flight control and can modify all configurable parameters in PX4. After setting up MAV_1_CONFIG= TELEM2, MAV_1_MODE = Onboard and SER_TEL2_BAUD =921600, MAVLink is available on the TELEM 2 port. However, we still need to set up TX2 to convert

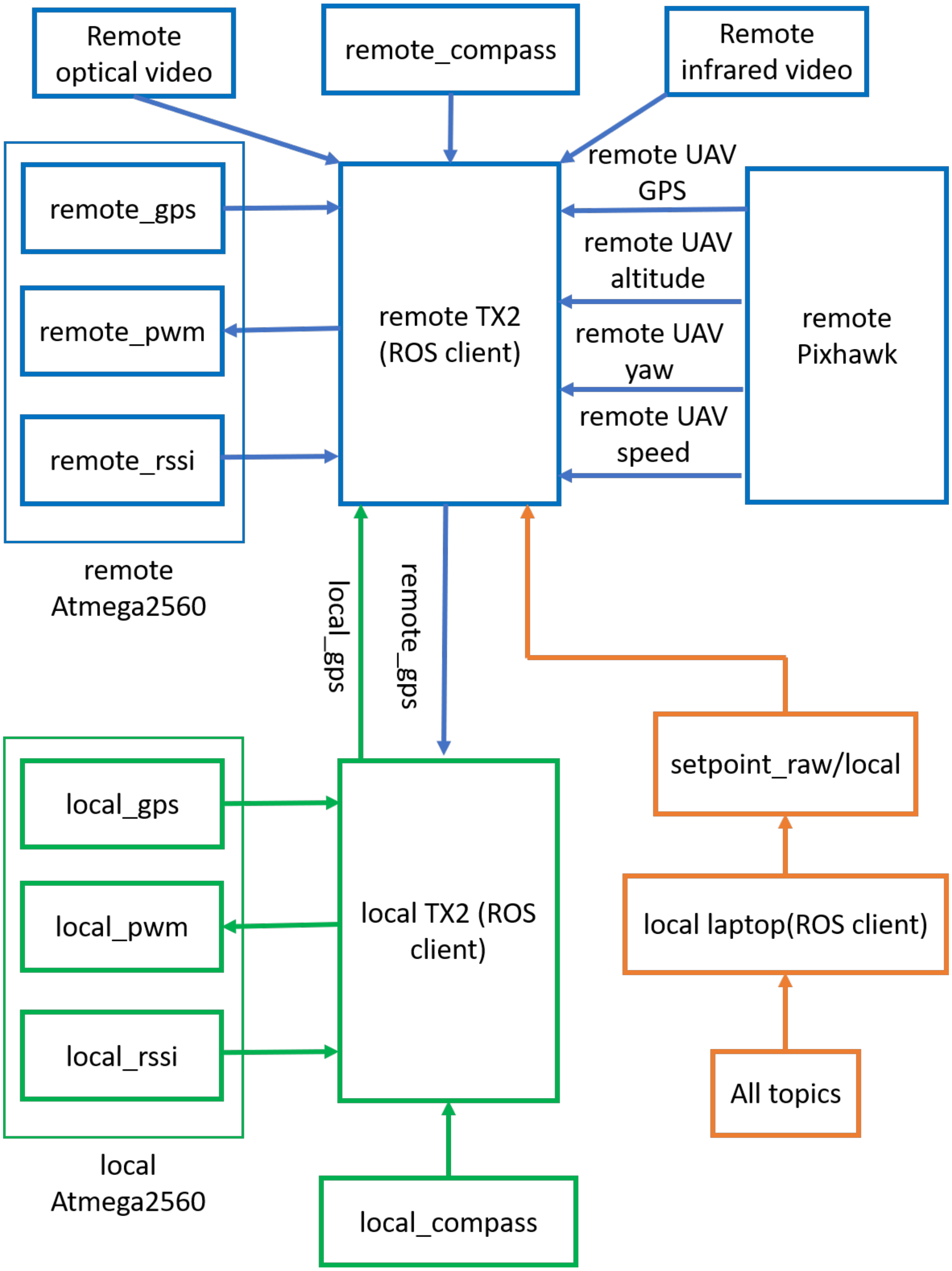


Figure 4.8. ROS architecture and topics in the ACDA system..

MAVLink messages to ROS messages so that a local laptop can completely control the remote UAV.

4.3.1.2 TX2 Setup

By utilizing ROS, we don't need to build a custom communication link between a local laptop and the remote Pixhawk. We only need to convert MAVLink messages to ROS topics. In fact, we can use MAVROS ROS package to build a bridge between MAVLink messages and ROS topics (see Figure 4.9). MAVROS publishes messages, such as the IMU state, local position in the North-East-Down (NED) coordinate, global position information, and system status, to the ACDA system. MAVROS subscribes the following ROS topics to control UAVs. ROS topic 'setpoint_raw/local' controls the local position, velocity and acceleration of a UAV in the local coordinate. ROS topic 'setpoint_raw/attitude' controls the attitude, angular rate and thrust of a UAV. In addition, MAVROS includes some services to control the UAV, such as waypoint service to set up waypoints, param service to access the parameters of PX4, and command services to send arm, takeoff, land commands to Pixhawk, and check UAV status.

After MAVROS is installed on TX2, run the command 'roslaunch MAVROS px4.launch fcu_url:=/dev/ttyUSB0:921600' in a terminal window of TX2. px4.launch is to execute ROS PX4 note to convert MAVLink messages to ROS topics. /dev/ttyUS B0 is the linked port of Pixhawk in TX2, and 921000 is the baud rate to communicate between TX2 and Pixhawk. The above command connects Pixhawk with the ACDA system.

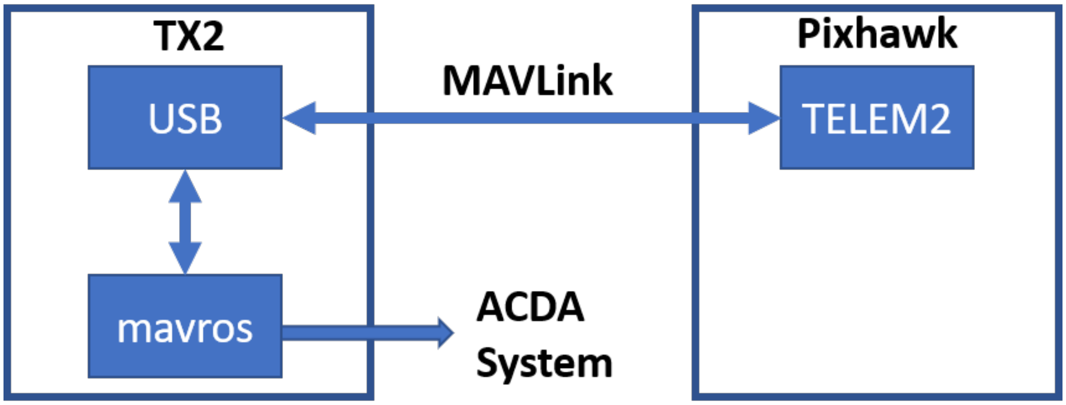


Figure 4.9. The process from MAVLink message to ROS Topics.

4.3.1.3 Teleoperation

In the MAVLink protocol, off-board control of PX4 flight can use two commands, SET_POSITION_TARGET_LOCAL_NED and SET_ATTITUDE_TARGET. Correspondingly in MAVROS, we can select the ROS topic 'setpoint_raw/local' and 'setpoint_raw/ attitude' for off-board control. In this chapter, we only use 'setpoint-_raw/local' for the BVLOS control of the remote UAV. The message type of 'setpoin_raw/local' is the mavros_msgs/PositionTarget which includes flight control parameters coordinate (NED or body coordinate), position, velocity, acceleration, yaw and yaw rate. We control the UAV with the velocity (VX, VY, VZ) and yaw_raw in the body coordinate of the UAV. We first select the body coordinate to do off-board control, and then assign ignore flag to position, acceleration and yaw parameters. As shown in Fig 4.10, the body coordinate in MAVROS is Right_Forward_Up (RFU). The body coordinate in MAVLink is Forward_Right_Up (FRU). Therefore, the velocity of 'setpoint_raw/local' shall be (VY, VX, VZ).

The keys w,x,a,d,q,e,z,and c in a standard keyboard are used to control the UAV's flight. The w and x control the UAV to move forward and backward. a and

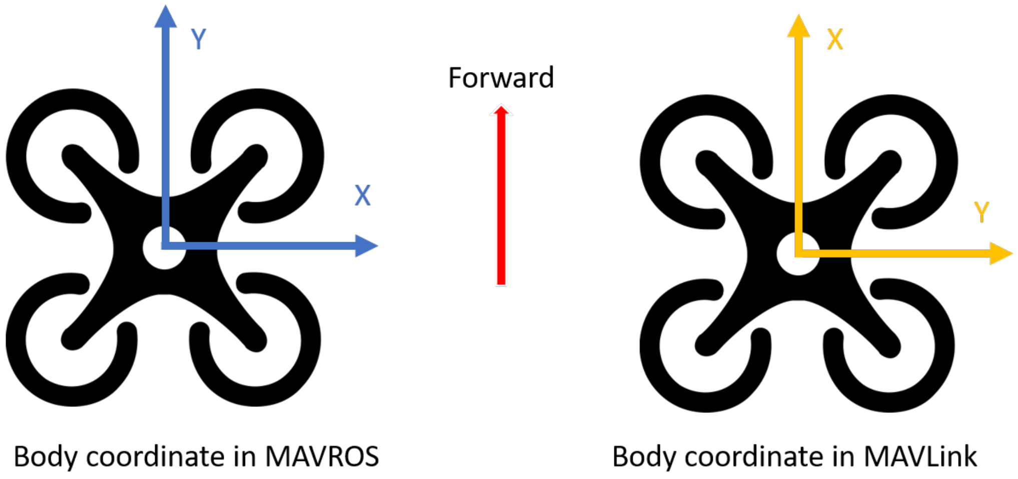


Figure 4.10. The Body Coordinate of the UAV in MAVROS and MAVLink.

d control the UAV to move left and right. The q and e control the UAV to ascend and descend. The z and c control the UAV to rotate clockwise and anticlockwise. The s lets velocities and yaw rate of the UAV be zero and hence control the UAV to hover in the air. In addition, if the time between two offboard commands exceeds 0.5 second, the flight mode of the PX4 will go out of the offboard model and restore the last mode of the Pixhawk. As such, the frequency of sending offboard commands must be larger than $2Hz$.

4.3.2 Simulation Studies

4.3.2.1 Simulation Environment

Before testing offboard control on a real UAV, it is useful to first simulate and debug. PX4 provides such a capability. According to [70], PX4 supports the Hardware In the Loop (HITL) simulation with the Gazebo simulator [71], which is a powerful simulation tool that is compatible with ROS. In order to simulate PX4 in Gazebo, PX4 firmware must be installed in the ROS work space on a laptop with the

Ubuntu operating system. PX4 Firmware in the Gazebo environment does not only simulate a UAV but also simulate multiple UAVs. In addition, they also simulate the data from all the sensors, including IMU, GPS, camera and barometer modules. They can even simulate the wind environment for the simulated UAVs and create noise to corrupt GPS signals.

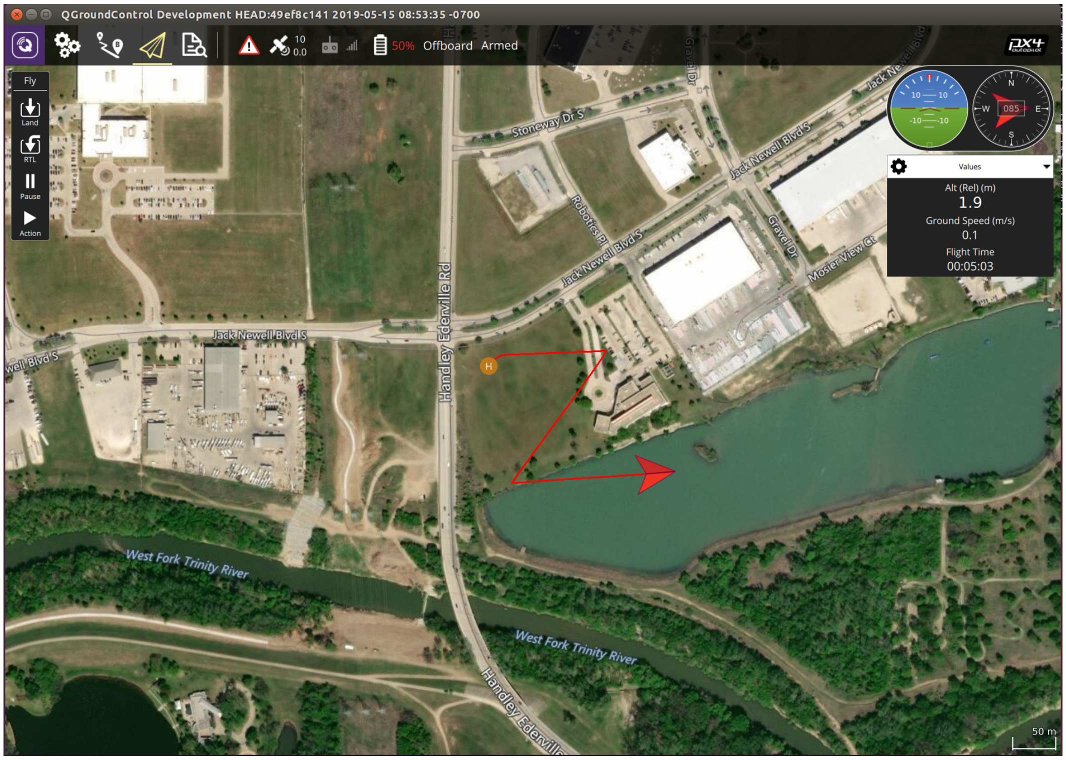
4.3.2.2 BVLOS UAV Control Simulation with Gazebo

The following procedures are involved. First, run the command 'make px4_sitl gazebo' in the PX4 firmware folder in a terminal window to create a Quadcopter in the Gazebo simulator. Second, to connect the simulated Quadcopter with ROS via MAVROS, execute the command 'roslanch MAVROS px4.launch fcu_rul:="udp://:14 540@127.0.0.1:14557"'. Becuase we simulate our offboard program on one laptop, the IP address of the laptop is the locallhost. An offboard API transmits message with PX4 using port 14540, which is for ROS initial connection. Third, open the QGC to monitor the Quadcopter movement and status. Finally, execute BVLOS UAV control. In this simulation, we make the Quadcopter move a 'Z' trajectory. The simualation result is shown in the Fig 4.11.

4.3.3 Field Tests

To test performance of the ACDA system, we let the local and remote UAVs fly at the same altitude, and then we measure the end-to-end communication performance between the local laptop and remote laptop by Iperf [72]. The throughput of the ACDA system is measured with the different distances as

Distance	1000 m	2000 m	4000 m
Throughput	45 Mbps	40 Mbps	20 Mbps



QGroundControl Development HEAD:49ef8c141 2019-05-15 08:53:35 -0700

50% Offboard Armed

Alt (Rel) (m) 1.9

Ground Speed (m/s) 0.1

Flight Time 00:05:03

Figure 4.11. BVLOS UAV Control in Gazebo using ROS .

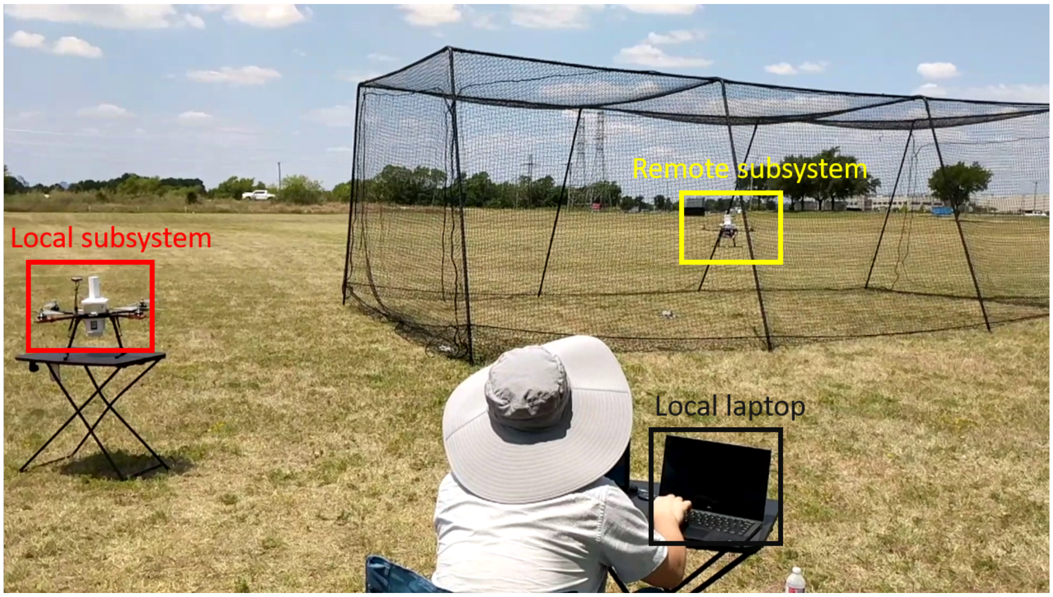
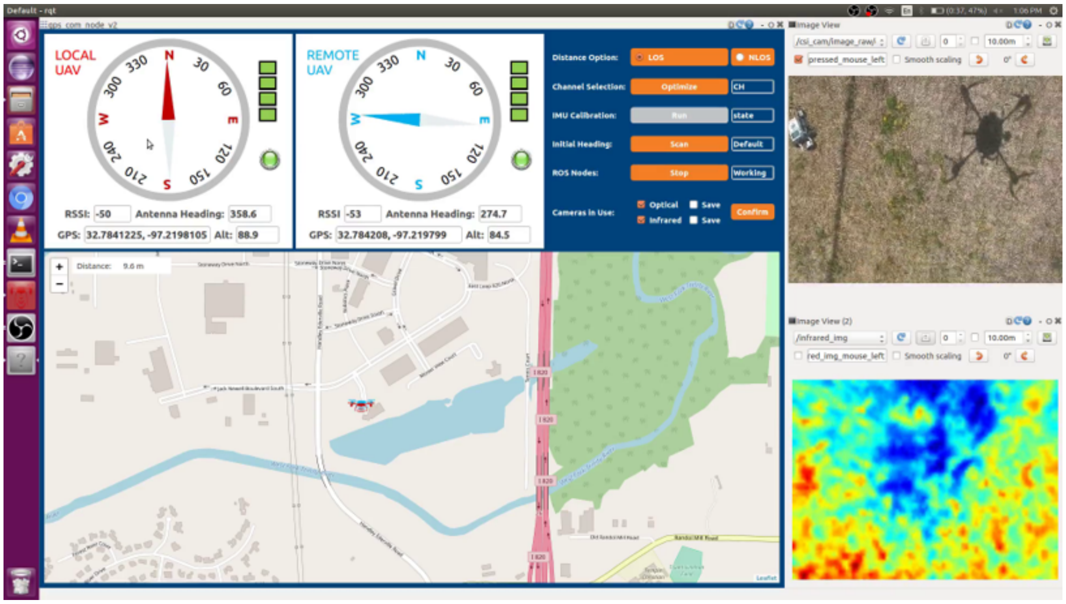


Figure 4.12. Test of the teleoperation in the ACDA system.



RSSI: -50 Antenna Heading: 358.6
GPS: 32.7841225, -97.2198105 Alt: 88.9

RSSI -53 Antenna Heading: 274.7
GPS: 32.784208, -97.219799 Alt: 84.5

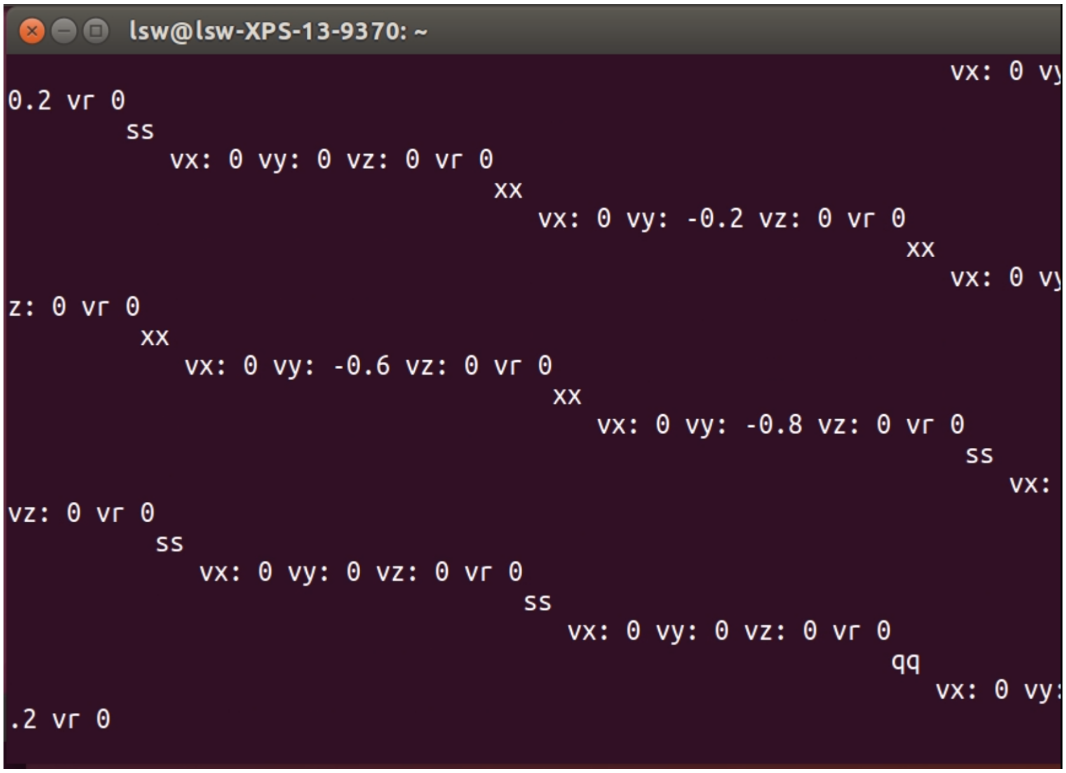
Distance Option: ● LOS ○ NLOS
Channel Selection: Optimize CH
IMU Calibration: Run state
Initial Heading: Scan Default
ROS Nodes: Stop Working
Cameras in Use: ☑ Optical □ Save ☑ Infrared □ Save Confirm

Distance: 9.6 m

/csi_cam/image_raw/ 0 10.00m
☑ pressed_mouse_left □ Smooth scaling 0°

/infrared_img 0 10.00m
□ red_img_mouse_left □ Smooth scaling 0°

Figure 4.13. View from the remote UAV displayed in the interface of the ACDA system.


lsw@lsw-XPS-13-9370: ~

```
                                                                vx: 0 vy
0.2 vr 0
        ss
           vx: 0 vy: 0 vz: 0 vr 0
                                 xx
                                    vx: 0 vy: -0.2 vz: 0 vr 0
                                                             xx
                                                                vx: 0 vy
z: 0 vr 0
         xx
            vx: 0 vy: -0.6 vz: 0 vr 0
                                     xx
                                        vx: 0 vy: -0.8 vz: 0 vr 0
                                                                 ss
                                                                    vx:
vz: 0 vr 0
          ss
             vx: 0 vy: 0 vz: 0 vr 0
                                   ss
                                      vx: 0 vy: 0 vz: 0 vr 0
                                                            qq
                                                               vx: 0 vy:
.2 vr 0
```

Figure 4.14. Illustration of the offboard flight commands.

CHAPTER 5

Design and Implementation of a Remote UAV-based Mobile Health Monitoring System

5.1 Introduction

Unmanned aerial vehicles as an easy-to-operate and cost-effective carrier of sensors have played increasing roles in nondestructive health monitoring [73, 74]. For instance, a UAV platform equipped with camera has been developed for bridge inspection [75]. A UAV carried laser scan sensing system has been used to check power lines [76]. In all existing UAV health monitoring applications, infrastructures of interest always have fixed geolocations, and hence the design focus of UAS monitoring system has been mostly on UAV stability and signal processing of monitoring data.

With growing mobility in modern Internet-of-Things (IoT) applications, the health monitoring of mobile infrastructures becomes an emerging need. UAVs, with their nice mobility features, provide a natural means to address this new health monitoring problem. Health monitoring for mobile infrastructures introduces two additional challenges. First, precise tracking of mobile infrastructures is needed in unfriendly outdoor environment. Many precise UAV tracking systems built for indoor environment were equipped with several cameras or lasers for precise positioning [77–79]. In outdoor environment where such positioning equipment does not exist, GPS signal is not stable, and disturbances (such as strong winds) disrupt UAV operations, a precise UAV tracking system becomes challenging. As such, a precise UAV tracking becomes challenging. Second, monitoring data may need to be transmitted over a long distance due to the movement of mobile infrastructures. However, existing commercial UAV communication systems only support the transmission of pictures/videos within a short distance, and do not meet the long-range operation need for mobile infrastructure health monitoring. In this chapter, we develop and prototype an outdoor and long-range mobile infrastructure health monitoring system, with novel tracking and communication system designs to address these new challenges. Next, we review related works on outdoor UAV tracking and UAV communication solution to motivate our design.

A tracking solution was developed to have a UAV locate a mobile target on the ground through the control of a fix-wing UAV's turn rate and gimbal's pan rate [80, 81]. The algorithm views the target as a point object, and relies on GPS and altitude measurement to maintain relative distance to the target. In order to maintain correct relative position without GPS measurement, paper [82] uses a set of features on the moving target for relevant position measurement. This tracking solution requires the UAV to be equipped with a high-precision camera and be very close to the target to differentiate these features. In paper [83], a UAV tracks a target based on a 2-dimensional picture using a fixed front camera, and hence does not estimate the depth nor maintain relative attitude. Different from the above works, we here develop a UAV tracking solution of the following features: 1) correct maintenance of relative position in three dimensions, 2) large view coverage to deal with sharp target trajectory changes, and 3) robust operation in GPS-denied environment, and 4) cost-effective solution that does not require high-performance cameras nor high-performance computing devices.

Along the direction of monitoring video stream transmission, existing health monitoring system solutions either store monitoring videos on-board and processes them later [80–83], or use standard Wi-Fi links for real-time data transmission. However, a typical Wi-Fi link only covers $100 - 200$ meters, which are not sufficient for remote monitoring of mobile infrastructures. To extend the communication distance, aerial networks can be used. Omni-directional aerial network-based solutions have been developed [84–86], which require multiple UAVs to be deployed for relay purposes. We have been developing directional antenna-based solutions to extend communication distance with a minimal number of UAVs deployed in the area [11, 12, 14, 19, 20, 63], which we adopt in this study.

In this chapter, we develop an effective UAV-carried vision-based infrastructure health monitoring system that allows a UAV to continuously track and monitor a mobile infrastructure and transmit back the monitoring information in real-time from a remote location. Based on a novel vision-based leader-follower tracking algorithm, the monitoring system uses a simple UAV-mounted and cost-effective camera and requires only a single feature on the mobile infrastructure and light computation for robust target detection and tracking. In addition, a UAV-carried aerial networking infrastructure using directional antennas is used to enable robust real-time transmission of monitoring video streams over a long distance. Automatic heading control is used to self-align headings of directional antennas to enable robust communication in mobility.

The rest of this chapter is organized as follows. Chapter 5.2 describes the system design with an emphasis on hardware components for tracking control and directional communication systems. Chapter 5.3 details a tracking control solution using one camera and one feature on the mobile infrastructure. Chapter 5.4 demonstrates the simulation and implementation results.

5.2 System Design

In this section, we first describe the overall architecture of the Remote UAV-based Mobile Health Monitoring System (RUMHMS). We then describe the main hardware components for both vision-based tracking and directional antenna heading control. Notations used in this chapter are summerized in the following table.

Notations used in this chapter are summerized in the following table.

$X_iY_iZ_i$	Inertial frame of a UAV	$X_bY_bZ_b$	Body frame of a UAV
$X_cY_cZ_c$	Camera frame	O_i	Origin in $X_iY_iZ_i$
ϕ, θ, ψ	Roll, pitch, yaw angles in $X_bY_bZ_b$	O_c	Center of a camera
δ	Pan angle of a camera's gimbal	η	Tilt angle of a camera's gimbal
$\mu\nu$	Image plane	r	Radius of a sphere-shaped target
$O_{\mu\nu}$	Center of image coordinates	(m_0, n_0)	Position of $O_{\mu\nu}$ in $\mu\nu$
P	Projection of the sphere-shaped target on $\mu\nu$	A	Area of P
O_B	Center of the target in $X_iY_iZ_i$	$O_{B_{\mu\nu}}$	Projection of O_B on $\mu\nu$
(m_B, n_B)	Position of $O_{B_{\mu\nu}}$ in $\mu\nu$	x_{cB}, y_{cB}, z_{cB}	Position of O_B in $X_cY_cZ_c$
(x_{iB}, y_{iB}, z_{iB})	Position of O_B in $X_iY_iZ_i$	x_{bB}, y_{bB}, z_{bB}	Position of O_B in $X_bY_bZ_b$
(x_c, y_c, z_c)	A point in $X_cY_cZ_c$	(m, n)	Projection of (x, y, z) on $\mu\nu$
f	Focal length of a camera	α_μ	Pixel dimensions at the μ axis
α_ν	Pixel dimensions at the ν axis	f_y	f/α_μ

f_y	f/α_ν	ξ	Bearing angle between X_z and O_cO_B
$\sum F$	The external forces on UAVs	m	The total mass of a UAV
ω	The UAV's angular velocity in $X_bY_bZ_b$	V	The translational velocity in $X_bY_bZ_b$
$\sum T$	The external moments of a UAV	I	The diagonal inertial matrix of a UAV
(e_{xi}, e_{yi}, e_{zi})	The position errors of a target in $X_iY_iZ_i$	e_{xb}, e_{yb}	The position errors at X_b and Y_b axises in $X_bY_bZ_b$
T_{hr}	The throttle size	$K_{p,\phi}$, $K_{d,\phi}$, $K_{i,\phi}$	The proportional gain, derivative gain, and integral gain in the roll control
$K_{p,\theta}$, $K_{d,\theta}$, $K_{i,\theta}$	The proportional gain, derivative gain, and integral gain in the pitch control	$K_{p,T_{hr}}$, $K_{d,T_{hr}}$, $K_{i,T_{hr}}$	The proportional gain, derivative gain, and integral gain in the throttle control
δ_{max}	The largest angle of δ	$e_{A_{ub}}$	The error between directions of UAV and the target
(e_m, e_n)	The target's position errors in the image plane	$\dot{\delta}$, $\dot{\eta}$	Rates of the pan and tilt angles

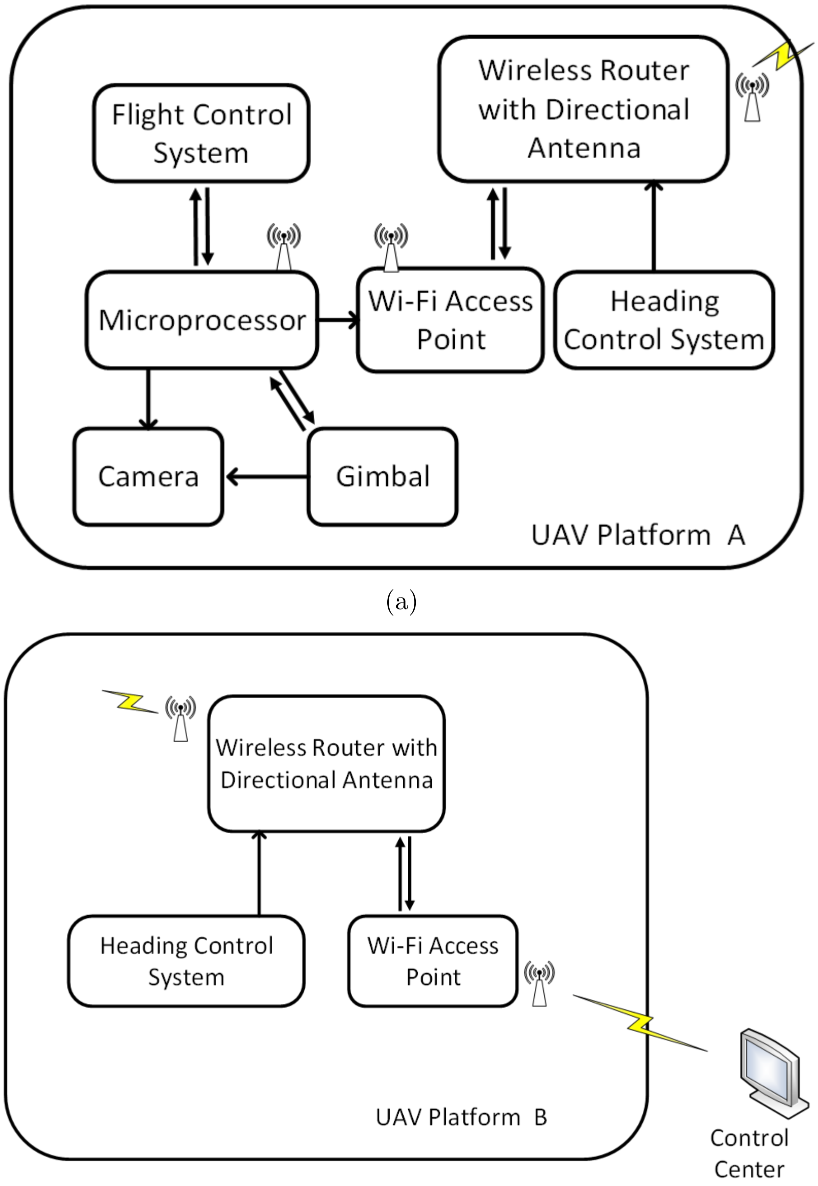
$K_{p,\delta}$	The proportional gain in the pan control	$K_{p,\eta}$	The proportional gain in the tilt control
$K_{p,\psi}$, $K_{d,\psi}$, $K_{i,\psi}$	The proportional gain, derivative gain, and integral gain in the yaw control	ϕ_d, ϕ_d, θ_d	The control outputs in the roll, pitch and throttle control
ψ_d	The control output in the yaw control		

Table 5.1: Notations

5.2.1 An Overview of RUMHMS

As shown in Figure 5.1, RUMHMS consists of platforms on two sites: one on the remote site over the mobile infrastructure (Platform A on the left) and one at the control center (Platform B on the right). Functionalities of Platform A are three-folds.

- *Image processing and relative position estimation.* A UAV-carried camera captures real-time videos of the mobile infrastructure (with a sphere-shaped object placed on top) and sends the video steam to a on-board microprocessor. The sphere-shaped target is one of key ideas of RUMHMS that allows using one feature to fulfill the tracking task. The microprocessor parses the video data to identify and locate the sphere-shaped target using color and shape identification functions in the openCV library. The parsed data including the location and area of the sphere in the image plane are used to estimate relative position of the target to the UAV [87].



(a)

(b)

Figure 5.1. The overall architecture of RUMHMS: (a) the architecture of UAV platform on the remote site over the mobile infrastructure, and (b) the UAV platform on the site of control center.

- *UAV and gimbal tracking control.* The microprocessor controls the UAV's pitch, roll and yaw angles and throttle to minimize the distance error between the desired and estimated relative positions for robust tracking under environmental disturbances. The attitude commands are then sent to the on-board autopilot for motor control. Meanwhile, the microprocessor controls the gimbal's pan and tilt servos to keep the target in the middle of the camera view.

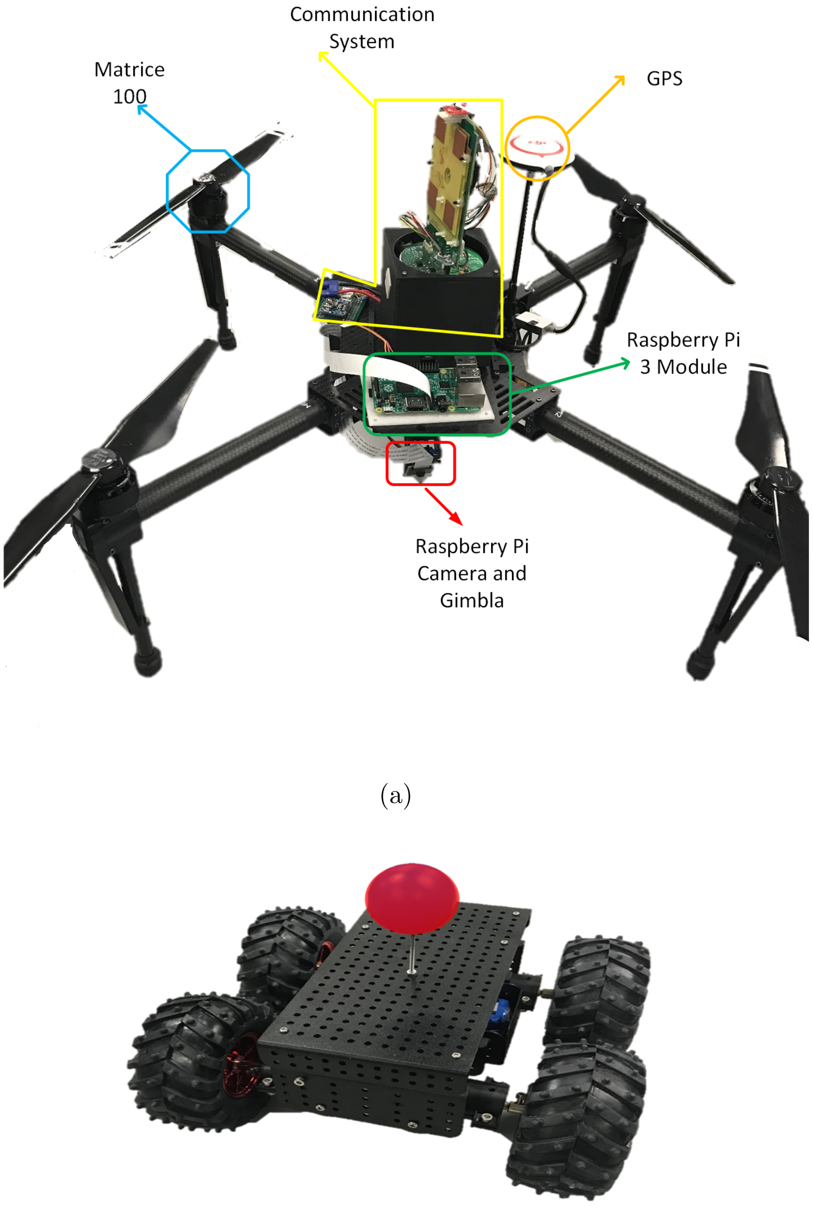
- *Directional video data transmission.* The monitoring video of the mobile infrastructure is sent to a control center via a directional wireless communication system to enable long-range communication. This directional communication system is enabled by two automatically aligned directional antennas on two UAVs, one at the site of mobile infrastructure and one at the control center. The alignment of directional antennas on two UAVs is enabled by the heading control system [11,12,14,20]. To facilitate the microprocessor and ground users to access RUMHMS without additional devices, we add a Wi-Fi access point on the UAV platform.

.

Platform B at the site of control center similarly consists of a Wi-Fi access point, directional antenna, and heading control system for directional communication. Experimental studies suggest that the throughput of directional air-to-air (A2A) communication link can reach 48Mbps at a distance of 300m and 2Mbps at a distance 5000m at 5Ghz [13]. An air to ground (A2G) is established in Platform B for the control center to access the remote monitoring video stream.

5.2.2 The Main Components of RUMHMS

In the section, we describe the hardware components that we choose for RUMHMS in our implementation. The prototype system is shown in Figure 5.2.



(a)

(b)

Figure 5.2. Hardware implementation: (a) the main components of the RUMHMS, and (b) the ground mobile infrastructure testbed with a sphere-shaped target for testing and validation purposes.

5.2.2.1 UAV Platform

UAV is a vital component for RUMHMS, because it carries the flight control system, camera, communication system, heading control system, batteries, and all other necessary devices. We consider the following factors when selecting a UAV platform: frame weight, takeoff weight, battery life, lift force, flexbility of software development kit (SDK), and stability in strong winds. With such considerations, we select DJI Matrice 100 carbon fiber quadcopter frame as our UAV platform[54], as shown in Figure 5.2(a). It weighs 2431g with batteries. Its takeoff weight achieves up to 3.6 kg. Two 5700 mAh lipo batteries can offer the power for over 40-minute flight time. Matrice 100 communicates with many operational systems, such as windows, linux and embedded systems, using a universal asynchronous receiver/transmitter(UART) interface. In addition, Matrice 100 offers a flexible onboard SDK, which allows microcontrollers to easily access and control UAV attitude and set way-points.

5.2.2.2 Microprocessor

We choose Raspberry Pi 3 (RP3)[88] as the microprocessor for image processing, and UAV and gimbal tracking control. This microprocessor has a 1.2GHz 64-bit quad-core ARMv8, 802.11n Wireless LAN, and is also cost-effective and light in weight. It can communicate with the autopilot of Matrice 100 via a UART port to control the UAV attitude. It can access the RUMHMS system with its built-in Wi-Fi dongle. In addition, RP3 has a pulse-width modulation (PWM) pinout for gimbal control.

5.2.2.3 Camera

We choose Raspberry pi camera module [89] for both vision-based tracking and health monitoring. Although this camera has a small board size ($25mm \times 25mm \times$

$25mm$), it supports 1080p30, 720p60 and 640x480p 60/90 video records. With Raspberry pi 3 module connected to the network, videos from the Raspberry Pi camera can be sent to computers and mobile devices which are in the same network.

5.2.2.4 Wireless Routers and Antennas

Wireless routers with directional antennas play a critical role for A2A communication in RUMHMS. Two directional antennas transmit video streams over a long distance. We choose Ubiquiti Nanostation LocoM5 for this purpose [90]. Its wireless router supports standard Wi-Fi access. Two paired Nanostation LocoM5s have a proprietary protocol. Messages can be transmitted through this protocol using directional antennas. Wireless access point supports ground to air and A2G communication, and all on-board devices (e.g., camera) to access RUMHMS. We select Huawei WS322 mini router to support $2.4/5Ghz$ Wi-Fi signals. It has low weight ($19.6g$) and compact size ($67mm \times 65mm \times 10mm$).

5.2.2.5 Heading Control

The heading control system aligns two directional antennas to enable high-throughput A2A communication. Components of this system include GPS, E-compass, Wi-Fi module, motor and microprocessor, as shown in Figure 5.3. The GPS module locates antenna position. E-Compass measures current antenna heading. A microprocessor Beaglebone Black (BBB) connects with GPS, E-compass and Wi-Fi module to control the antenna heading and send the GPS data to the other UAV platform through the directional communication system. The Wi-Fi module provides microprocessor the access to RUMHMS. Received signal strength indicator from the other directional antenna's Wi-Fi router is used for heading control in denied GPS environment [14, 20].

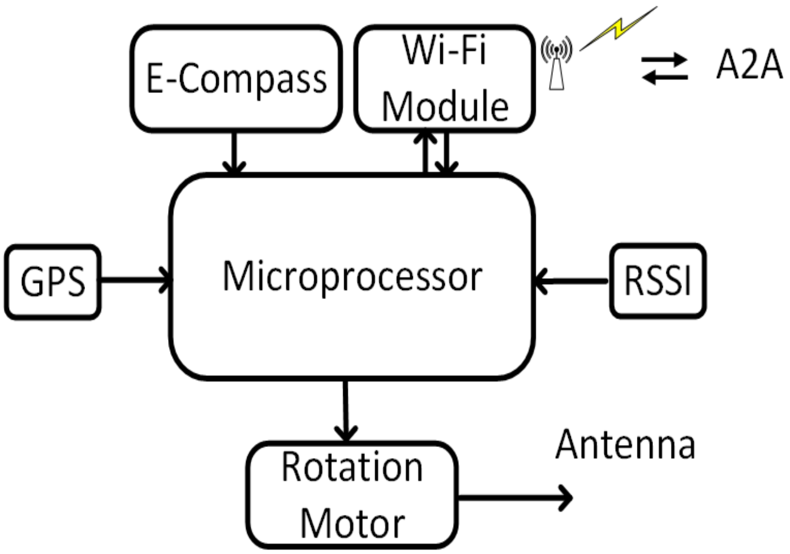


Figure 5.3. The heading control module.

5.3 UAV and Gimbal Tracking Control

In this section, we first present a method to estimate a sphere-shaped target's relative location in the UAV's inertial frame using a single camera carried by the UAV. We then present an overview of UAV's kinematic model. In the end, a vision-based control law is developed for the UAV and gimbal to together track the mobile target.

5.3.1 Relative Target Location Estimation using a Single Camera

To find the sphere-shaped target's relative location in the UAV inertial frame $X_iY_iZ_i$ (North-West-Up), we first locate the target in the camera frame, and then transfer it to the body frame and finally the inertial frame. As shown in Figure 5.4, a sphere with radium r projects an ellipse P with center $O_{B_{\mu\nu}}$ onto the image plane $\mu\nu$.

We first locate the sphere's position in the camera frame $X_cY_cZ_c$ using A and the position (m_B, n_B) of $O_{B_{\mu\nu}}$ measured on the image plane and also the sphere's

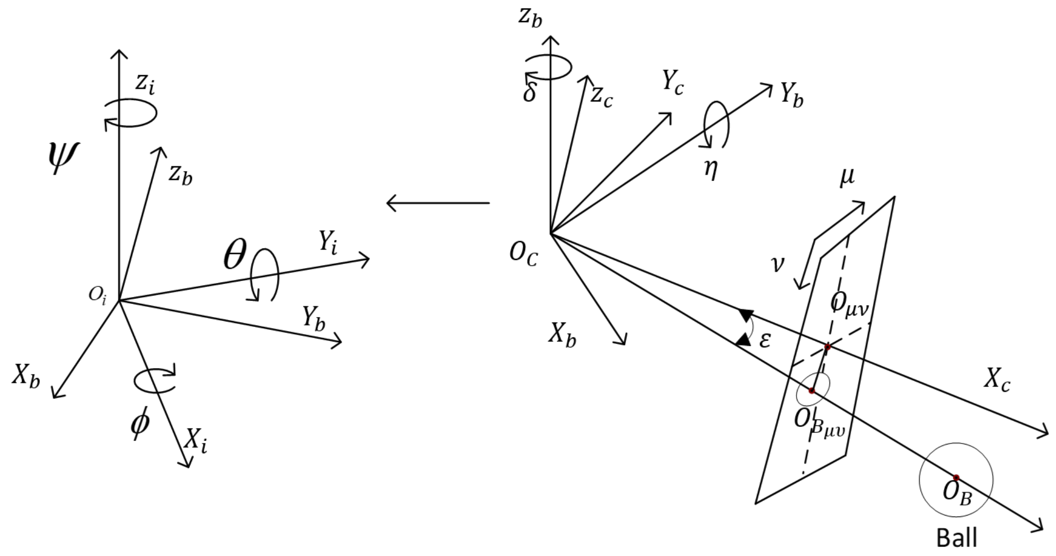


Figure 5.4. Coordinates for location.

radius r. We use (x_{cB}, y_{cB}, z_{cB}) to denote the sphere's position in $X_cY_cZ_c$. According to the camera projection model[91], we have

$$\begin{aligned} m &= f_y \tfrac{y_c}{x_c} + m_0 \\ n &= f_z \tfrac{y_c}{x_c} + n_0 \end{aligned} \tag{5.1}$$

where (x_c, y_c, z_c) is an arbitrary point in $X_cY_cZ_c$, (m_0, n_0) is the center of the image plane, (m, n) is the projection of (x_c, y_c, z_c) on the image plane $\mu\nu$, $f_y = f/\alpha_\mu$, $f_z = f/\alpha_\nu$, f is the focal length of a camera, α_μ is the pixel dimensions at the μ axis, and α_ν is the pixel dimensions at the ν axis. The projected location relationship of the sphere in the camera frame and the image plane can be described as

$$\begin{aligned} \tfrac{y_{cB}}{x_{cB}} &= \tfrac{m_B - n_0}{f_y} \\ \tfrac{z_{cB}}{x_{cB}} &= \tfrac{n_B - n_0}{f_z} \end{aligned} \tag{5.2}$$

In addition, x_{cB} and angle ξ, the bearing angle between X_c axis and the line $O_cO_{B_{\mu\nu}}$ can be described as [87]

$$x_{cB} = r\sqrt{\pi/(Acos\xi)} \tag{5.3}$$

$$\xi = \arctan\sqrt{\frac{y_{cB}^2}{x_{cB}^2} + \frac{z_{cB}^2}{x_{cB}^2}} \tag{5.4}$$

We then calculate y_{cB} and z_{cB} using the following equations

$$\begin{aligned} y_{cB} &= x_{cB}(\tfrac{m_B - n_0}{f_y}) \\ z_{cB} &= x_{cB}(\tfrac{n_B - n_0}{f_z}) \end{aligned} \tag{5.5}$$

To locate the sphere's location in $X_iY_iZ_i$, we need to transfer the sphere's position (x_{cB}, y_{cB}, z_{cB}) from the camera frame $X_cY_cZ_c$ to the body frame $X_bY_bZ_b$ and then to the inertial frame $X_iY_iZ_i$. We assume the gimbal's frame is same as the camera's frame. In addition, we assume X_c is the optical axis of the camera. The rotation matrix R_c^b from the camera frame to the body frame can be described as

$$R_c^b = \begin{bmatrix} \cos\delta & \sin\delta & 0 \\ -\sin\delta & \cos\delta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\eta & 0 & -\sin\eta \\ 0 & 1 & 0 \\ \sin\eta & 0 & \cos\eta \end{bmatrix} \tag{5.6}$$

where δ is a pan angle of a camera's gimbal, η is a tilt angle of a camera's gimbal. The rotation matrix R_b^i from the body frame to the inertial frame is described as

$$R_b^i = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{5.7}$$

where ϕ, θ, ψ are roll angle, pitch angle and yaw angle in the body frame. In our experimental set-up, the camera (with gimbal) is installed under the central bottom of a UAV, and hence the origin O_i in the inertial frame and the camera's center O_c are assumed to be the same point. Therefore, the position (x_{iB}, y_{iB}, z_{iB}) of O_B in the inertial frame $X_iY_iZ_i$ can be described as

$$\begin{bmatrix} x_{iB} & y_{iB} & z_{iB} \end{bmatrix}^T = R_b^i R_c^b \begin{bmatrix} x_{cB} & y_{cB} & z_{cB} \end{bmatrix}^T \tag{5.8}$$

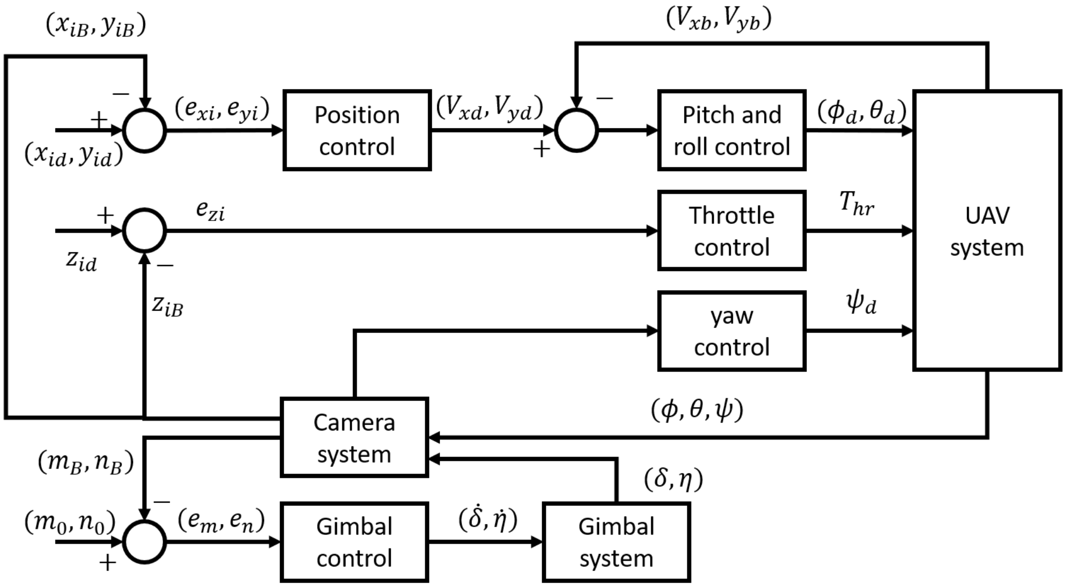


Figure 5.5. Control diagram for UAV and gimbal.

5.3.2 UAV Kinematics

We use Matlab simulink [92] for the model of UAV kinematics. Briefly, the UAVs' dynamical equations are described as follows according to the Newton-Euler formalism [93]

$$\begin{aligned} \sum F &= m\dot{V} + \omega \times V \\ \sum T &= I\dot{\omega} + \omega \times I\omega \end{aligned} \tag{5.9}$$

where the term $\sum F$ represents the external forces on UAVs, m is the total mass of a UAV, ω is the UAV's angular velocity in $X_bY_bZ_b$, V is the UAV's translational velocity in $X_bY_bZ_b$, $\sum T$ is the external moments of a UAV, and I is the diagonal inertial matrix of a UAV. More details can be found in the book[93] .

5.3.3 Control Laws for UAV and Gimbal

We build control laws for the UAV and gimbal to ensure a UAV to follow a sphere-shaped target on the mobile infrastructure (see Figure 5.5). In particular, the goal of control is to minimize the relative errors (e_{xi}, e_{yi}, e_{zi}) between the desired target position (x_{id}, y_{id}, z_{id}) and the current target position (x_{iB}, y_{iB}, z_{iB}) in the inertial frame $X_iY_iZ_i$ as follows:

$$\begin{aligned} e_{xi} &= x_{id} - x_{iB} \\ e_{yi} &= y_{id} - y_{iB} \\ e_{zi} &= z_{id} - z_{iB} \end{aligned} \tag{5.10}$$

To control a UAV's velocities at X_b-axis and Y_b-axis, the errors e_{xi} and e_{yi} need to be transferred from the inertial frame to the body frame using following equations

$$\begin{bmatrix} e_{xb} \\ e_{yb} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} e_{xi} \\ e_{yi} \end{bmatrix} \tag{5.11}$$

where e_{xb} and e_{yb} are position errors at X_b-axis and Y_b-axis in $X_bY_bZ_b$. We build P controllers to control the velocities along X_b-axis and Y_b-axis. The position controllers can be described as

$$\begin{aligned} V_{xd} &= K_{p,X_b} e_{xb} \\ V_{yd} &= K_{p,Y_b} e_{xb} \end{aligned} \tag{5.12}$$

where V_{xd} and V_{yd} are desired velocities along X_b-axis and Y_b-axis, K_{p,X_b} and K_{p,Y_b} are the proportional gains. Denoting the current velocities along X_b-axis and Y_b-axis as V_{xb} and V_{yb}, the velocity errors are

$$\begin{aligned} e_{V_{xb}} &= V_{xd} - V_{xb} \\ e_{V_{yb}} &= V_{yd} - V_{yb} \end{aligned} \tag{5.13}$$

We built PID controllers to control the roll angle ϕ_d, pitch angle θ_d and throttle size T_{hr} as follows

$$\begin{aligned} \phi_d &= K_{p,\phi} e_{V_{yb}} + K_{d,\phi} \dot{e_{V_{yb}}} + K_{i,\phi} \int e_{V_{yb}} dt \\ \theta_d &= K_{p,\theta} e_{V_{xb}} + K_{d,\theta} \dot{e_{V_{xb}}} + K_{i,\theta} \int e_{V_{xb}} dt \\ T_{hr} &= K_{p,T_{hr}} e_{zi} + K_{d,T_{hr}} \dot{e_{zi}} + K_{i,T_{hr}} \int e_{zi} dt \end{aligned} \tag{5.14}$$

where ϕ_d, θ_d, T_{hr} are the control outputs, $K_{p,\phi}$, $K_{d,\phi}$, and $K_{i,\phi}$ are the proportional gain, derivative gain, and integral gain in the roll control; $K_{p,\theta}$, $K_{d,\theta}$, and $K_{i,\theta}$ are the proportional gain, derivative gain, and integral gain in the pitch control; and $K_{p,T_{hr}}$, $K_{d,T_{hr}}$, and $K_{i,T_{hr}}$ are the proportional gain, derivative gain, and integral gain in the throttle control.

In this leader-follower tracking system, the relative position between the UAV and sphere is measured by the camera. As the camera's view is limited, we build pan and tilt controls for the gimbal to keep the sphere in the middle of the camera view. We build gimbal and yaw controls using the sphere's position errors (e_m, e_n) in the image plane to have the camera track the mobile sphere. More specifically, e_m and e_n are described as

$$\begin{aligned} e_m &= m_0 - m_B \\ e_n &= n_B - n_0 \end{aligned} \tag{5.15}$$

We use P controllers to control the gimbal's movement along the pan and tilt directions. The gimbal controllers are

$$\begin{aligned} \dot{\delta} &= K_{p,\delta} e_m \\ \dot{\eta} &= K_{p,\eta} e_n \end{aligned} \tag{5.16}$$

where $\dot{\delta}$ and $\dot{\eta}$ are rates of the pan and tilt angles, $K_{p,\delta}$ is the proportional gain for the pan control, and $K_{p,\eta}$ is the proportional gain for the tilt control. These controllers aim to keep the sphere in the center of the camera view.

As the movement of gimbal is limited in the pan and tilt directions, we also develop the yaw control. To simplify the yaw controller design, we assume here that the desired position of the sphere is $(x_{id}, 0, z_{id})$. The yaw control is

$$\psi_d = K_{p,\psi}\delta + K_{d,\psi}\dot{\delta} + K_{i,\psi}\int \delta dt \tag{5.17}$$

where ψ_d is the control output, and $K_{p,\psi}$, $K_{d,\psi}$, and $K_{i,\psi}$ are the proportional gain, derivative gain, and integral gain for the yaw control. We note that in this case the pan angle δ is the error between the desired and current yaw angles.

5.4 System Performance Evaluation

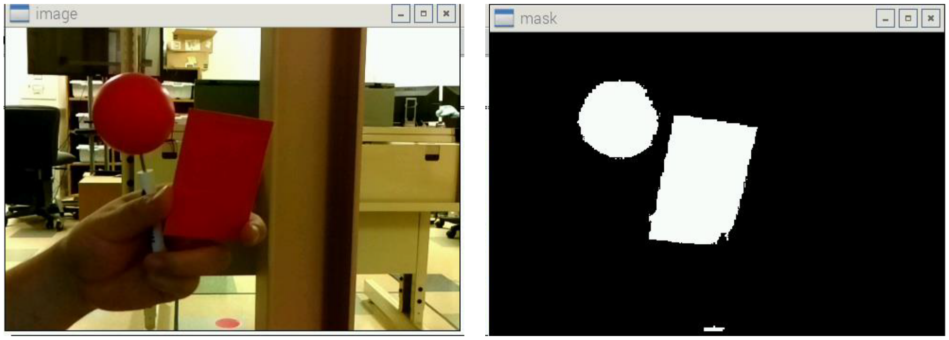
This section shows some of the simulation and testbed implementation results of RUMHMS. Initial testing and evaluation studies of RUMHMS successfully validated the design.

5.4.1 Target Detection Performance

Target detection is implemented using color and shape identification tools in the OpenCV library. First, Raspberry Pi 3 module (RP3) receives the detected target's image from the Raspberry pi camera. This color space of the image is transformed from the RGB (red, green, blue) cylindrical coordinates to HSV (hue, saturation, value) cylindrical coordinates. We use "red" in this implementation and its HSV range is from $[105, 65, 65]$ to $[125, 255, 255]$.

After color identification, the color image is transformed to a black-and-white image. The detected targets of the designated color are marked in white (Figure 5.6). The edges of white blocks are irregular as shown in Figure 5.6. To reduce the effect of noise on the measurement of positions and areas of detected targets, we apply Median Filtering in the OpenCV library to smoothen edges of these blocks as shown in Figure 5.7(a).

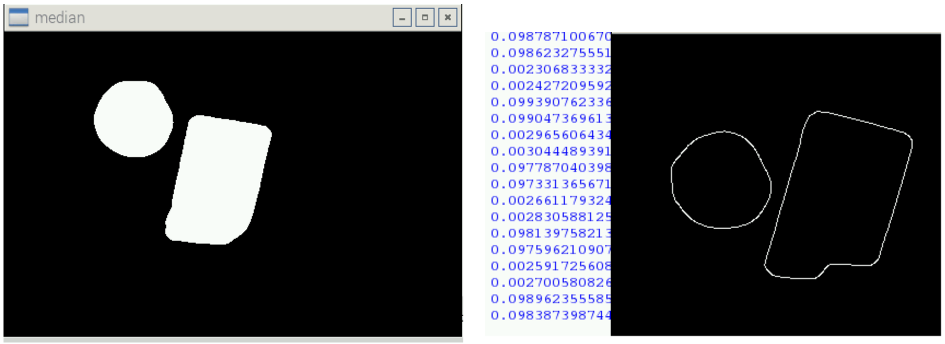
Finally, the contours of the white blocks are extracted to identify the detected shapes in the OpenCV library. We use sphere in this implementation. For each of detected contours, their dissimilarity ratios with a circle are calculated. A dissimilarity ratio of 0.01 is used to distinguish a sphere from other shapes of objects, as shown in Figure 5.7(b). Using the algorithm of image moments in the OpenCV library, the position and area of a detected contour are calculated and sent to Equations (2) and (3).



(a)

(b)

Figure 5.6. Illustration of color detection: (a) original objects, and (b) detected targets of the designated color.



```
0.09878710067O
0.098623275551
0.002306833332
0.002427209592
0.099390762336
0.099047369613
0.002965606434
0.003044489391
0.097787040398
0.097331365671
0.002661179324
0.002830588125
0.098139758213
0.097596210907
0.002591725608
0.002700580826
0.098962355585
0.098387398744
```

(a)

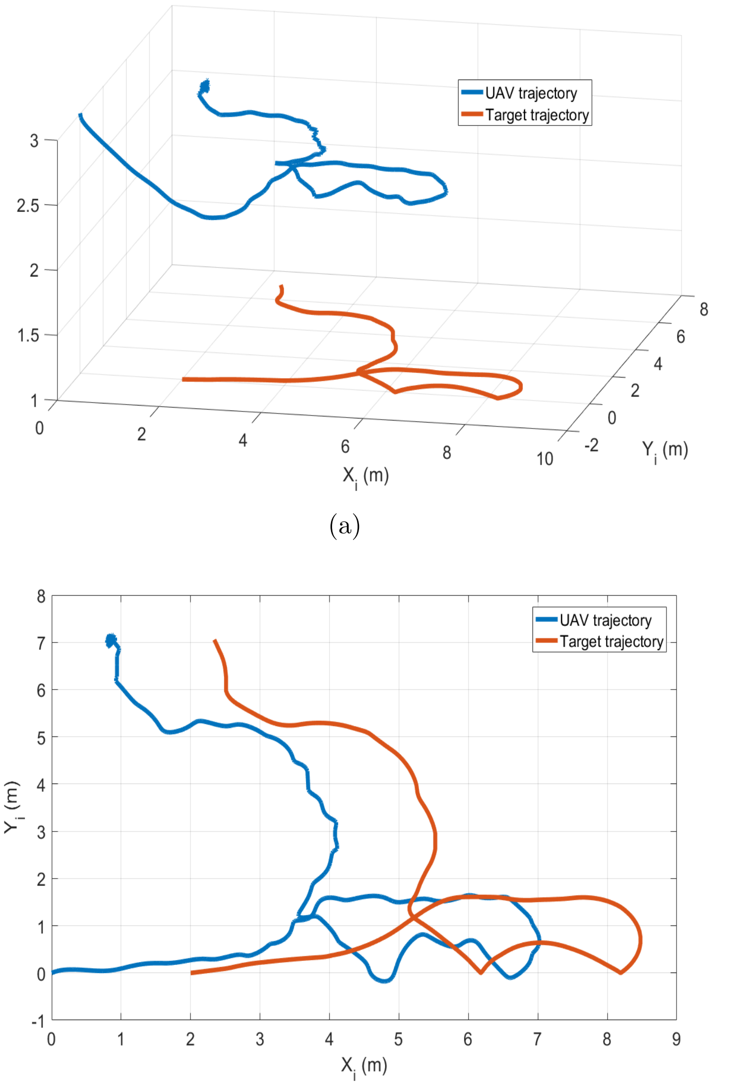
(b)

Figure 5.7. Illustration of shape detection: (*a*) smoothed edges of detected blocks, (*b*) contour extraction and shape recognition.

5.4.2 Tracking Performance

In simulation studies, we assume that the desired target's relative position (x_{id}, y_{id}, z_{id}) is $(1.5, 0, -1, 5)$. A UAV hovers at the initial position $(0, 0, 3)$ with the yaw angle 0 in $X_iY_iZ_i$. A sphere's initial position (x_{iB}, y_{iB}, z_{iB}) is $(2, 2, 3)$. The following parameters are used: $K_{p,X_b} = 1.25$, $K_{p,Y_b} = 1.25$, $K_{p,\phi} = 3.0$, $K_{d,\phi} = 1.2$, $K_{i,\phi} = 1.5$, $K_{p,\theta} = 3.5$, $K_{d,\theta} = 1.8$, $K_{i,\theta} = 2.5$, $K_{p,T_{hr}} = 3$, $K_{d,T_{hr}} = 4$, $K_{i,T_{hr}} = 1.5$, $m_0 = 420.7956$, $n_0 = 291.6529$, $K_{p,\delta} = 0.5$, $K_{p,\eta} = 0.2$, $K_{p,\psi} = 4$, $K_{d,\psi} = 3.5$, $K_{i,\psi} = 0.5$, $\sigma \in [-2/3\pi, 2/3\pi]$, and $\eta \in [-\pi/2, \pi/6]$. To mimic the movement of a target, its trajectory is generated from the smooth-turn mobility model [24] with total simulation time $100s$ and sampling time $1/350s$. A camera installed on the UAV provides measurement data to calculate relative position of the target in the inertial frame using Equations (1)-(8). The UAV controls its attitude using controllers (see Equations (12), (14), and (17)). Meanwhile, the camera controls its attitude to track the target using the controller in Equation (16).

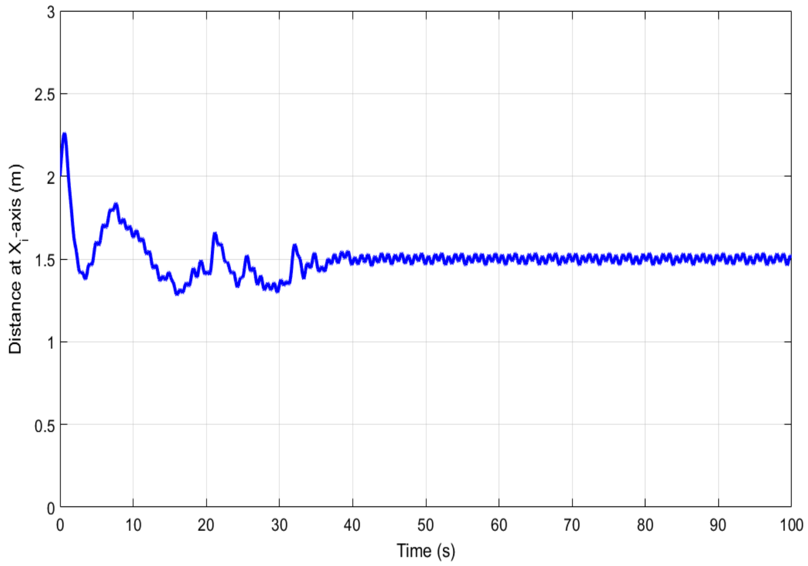
The trajectories of the UAV and target are shown in the Figure 5.8. Figure 5.8(a) illustrates the trajectories of the UAV and target in three dimensions. Figure 5.8(b) illustrates the trajectories of the UAV and target in the X_iY_i plane. Figure 5.9 illustrates the relative distances between the UAV and target in the inertial frame. The target's position is very close to the center of the image plane as shown in Figure 5.10. Simulation studies illustrate that the UAV can track the target at a desired relative position and keep the target at the center of the camera view. Implementation studies also show that the UAV can track a slowly moving vehicle well with tuned control parameters. More advanced adaptive controller designs and image processing techniques will be explored in the future works to improve the tracking performance.



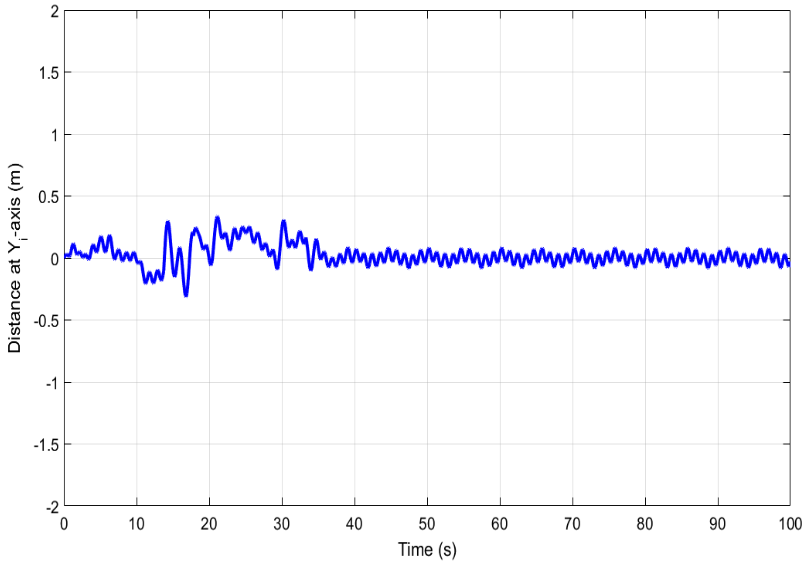
(a)

(b)

Figure 5.8. Trajectories of the UAV and target: (a) in three dimensions, and (b) in the X_iY_i plane.

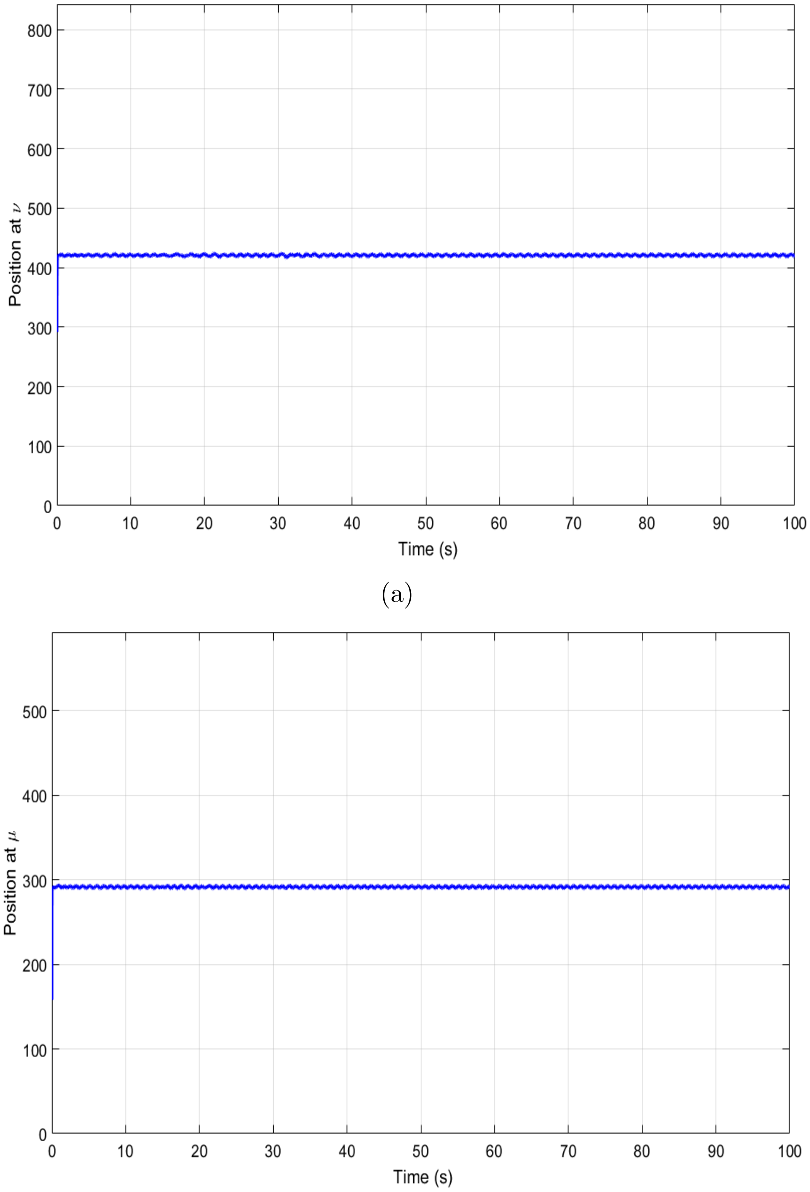


(a)



(b)

Figure 5.9. Relative distances between the UAV and target: (a) at X_i-axis, (b) at Y_i-axis, and (c) at Z_i-axis.



(a)

(b)

Figure 5.10. Positions of the target in the image plane: (a) at ν-axis, and (b) at μ-axis.

CHAPTER 6

Leader-follower Tracking Based on the Directional Antennas System

6.1 Introduction

Recently, more and more attention is being given to the applications of cooperative multi-UAVs. Some applications of cooperative multi-UAVs are implemented. For example, multi-UAV systems have been designed for agriculture [94, 95], remote sensing [8, 9], and monitoring [10, 96, 97] applications. Compared to single UAV applications [98–100], multi-UAVs are more effective in fulfilling time-critical missions that require a large area coverage. The control algorithms of flight formation have been well studied. Paper [101] presents a consensus-based cooperative control solution to enable multi-UAVs fly in formation with a leader-follower structure. The authors apply a collision-avoidance strategy based on an artificial potential approach for formation. Paper [102] proposes a sliding mode control for multi-UAV formation. In [103], a proportional-derivative (PD) control is implemented for the leader-follower formation. Paper [104] designs a multi-UAV formation using nonlinear model predictive control. In [105], the multi-UAV formation is implemented based on an artificial bee colony algorithm.

For flight formation, robust UAV-to-UAV communication is vital to transmit cooperative control signals, sensor data, and application-oriented video streams between multiple UAVs or UAVs and ground station. In [85, 102, 103, 106, 107], the communication system is built using omni-directional antennas or Wi-Fi routers. The aerial communication distance is only a few hundred meters, which is not sufficient for many practical UAVs applications. The above communication solutions limit the flexibility and applicability of multi-UAV formation.

To address this issue, we implement flight formation with a leader-follower structure using the ACDA communication system developed in [11, 12, 19–21]. The ACDA system supports long-distance and broadband robust communication. The aerial link of the ACDA system is established using directional antennas carried on UAVs. Several automatic alignment algorithms of directional antennas have been developed for robust communication performance. The local subsystem of the ACDA system works as a relay note to extend the communication distance between ground station and the remote UAV to kilometers. The Wi-Fi routers in the ACDA system provides wireless access so that ground devices can connect to the ACDA system without ground infrastructure support.

In this study, we develop a leader-follower tracking system for flight formation using the ACDA system. We first upgrade the ACDA system upon the version of the ACDA system in Chapter 4. Two flight control modules are connected to the ACDA system. The sensor data and flight status of two UAVs are converted to ROS topics using MAVROS with TX2. Meanwhile, two UAVs can subscribe to flight control commands from their companion computer TX2. Then, we develop and implement a cooperative multi-UAV system with a leader-follower pattern. A sliding mode control is adopted to automatically drive the local UAV to achieve at the desired position with respect to the remote UAV.

The rest of this chapter is organized as followers. In Chapter 6.2, we illustrate the upgraded ACDA system. Chapter 6.3 describes the leader-follower tracking controller design. Chapter 6.4 verifies the tracking control system using simulation studies.

6.2 Upgraded ACDA System Design

In this section, we first summarize the ACDA system we use for this study. Then we illustrate improvements on the ACDA system for leader-follower tracking.

6.2.1 Overview of the ACDA System

The ACDA system was designed and upgraded as described in [11], [20], Chapter 3 and Chapter 4 to enhance the performance. As shown in Figure 4.3, the subsystem of the ACDA system is composed of four parts, including computing, control, communication, and UAV platform. The computing component is realized using a microprocessor TX2, which runs ROS nodes to obtain sensors data, communicates with Pixhawk, and calculates the control signals of the directional antenna motor using a LQG controller for robust UAV-to-UAV communication. The control signals are sent to the control part for the automatic alignment of the directional antennas. The control part contains a rotating motor, a turnable plate, a motor driver, an ATmega2560, and a compass module. In addition, a lazy susan bearing is adopted in the turnable plate to enhance the rotational performance of the directional assembly. Based on the desired control signals, the ATmega2560 outputs stable PWM signals to the motor driver to rotate the directional antenna assembly. The communication component includes a directional antenna, a Wi-Fi router, and a Wi-Fi adapter on the TX2. This part is used to establish communication among the local system, remote system, and ground laptops. As shown in Figure 4.4, the air-to-air link is built using two directional antennas Nanostation Loco M5. The air-to-ground or ground-to-air link is built using Wi-Fi router Huawei WS323 with user devices. The air-to-computing or computing-to-Air link is built between the Wi-Fi router and the Wi-Fi adapter. The UAV platform is a Tarot 650 with a flight control module Pixhawk. The flight control signal of the Pixhawk works on $72Mhz$ which avoids the mutual inference with the Wi-Fi router ($2.4Ghz$). In addition, we can use a local laptop to subscribe to all ROS topics and monitor the status of the ACDA system.

6.2.2 The New ACDA System Design

For the leader-follower tracking system, it is necessary for the follower UAV to obtain the sensor data and status of the leader UAV. Since each of the local and remote UAVs is a leader UAV in the ACDA system, we need to connect two UAVs to the ACDA system. In Chapter 4, the remote flight controller module is connected into the ACDA system. The TELEM2 port of Pixhawk links with a USB port of TX2. Meanwhile, TX2 launches MAVROS to convert the MAVLink messages of the Pixhawk to ROS topics. Then, these ROS topics are published to the ROS network using TX2. The local Pixhawk is connected to the ACDA system using the same procedure with the remote Pixhawk. Compared to Figure 4.8, we connect the local flight controller module into the ACDA system (see Figure 6.1). The remote TX2 obtains the altitude, GPS, speed and yaw messages of the remote Pixhawk and publishes them to the ROS network. The local TX2 also obtains the same messages from the local Pixhawk. In the leader-follower tracking system, the TX2 in the follower UAV subscribes to ROS topics of the leader UAV and publishes the ROS topic 'setpoint_raw/local' to control the Pixhawk in the follower UAV. The ROS topic 'setpoint_raw/local' can control position, velocity, acceleration, yaw, and yaw rate of a UAV. In this work, we only control the velocities of 'setpoint_raw/local' in the local body North-East-Up (NEU) coordinate to achieve the leader-follower tracking.

6.3 Leader-follower Tracking Controller Design

We have designed and implemented the new ACDA system for the application of the leader-follower tracking system following [102]. In this subsection, we intro-

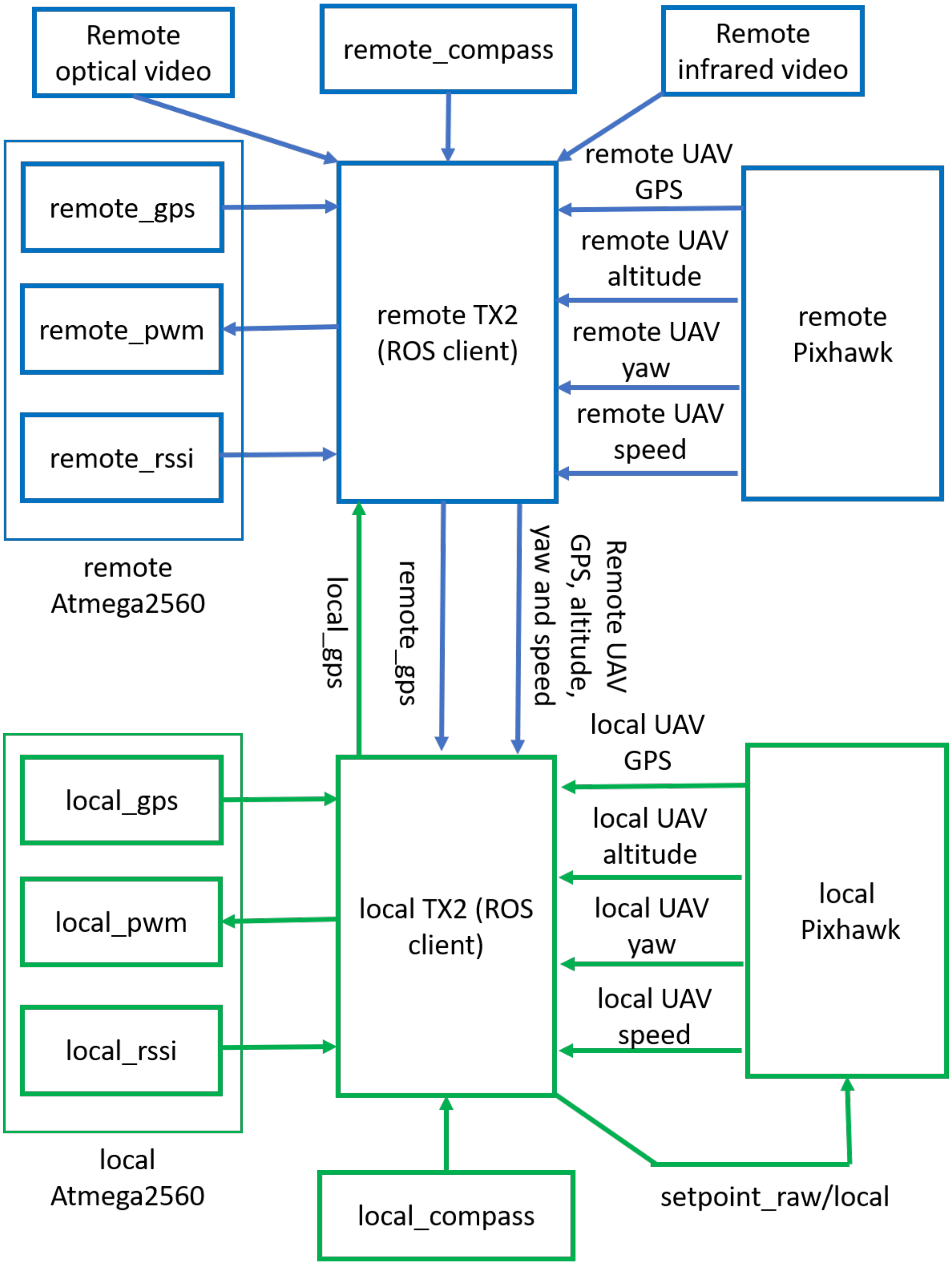


Figure 6.1. The ROS architecture and topics in the leader-follower ACDA system.

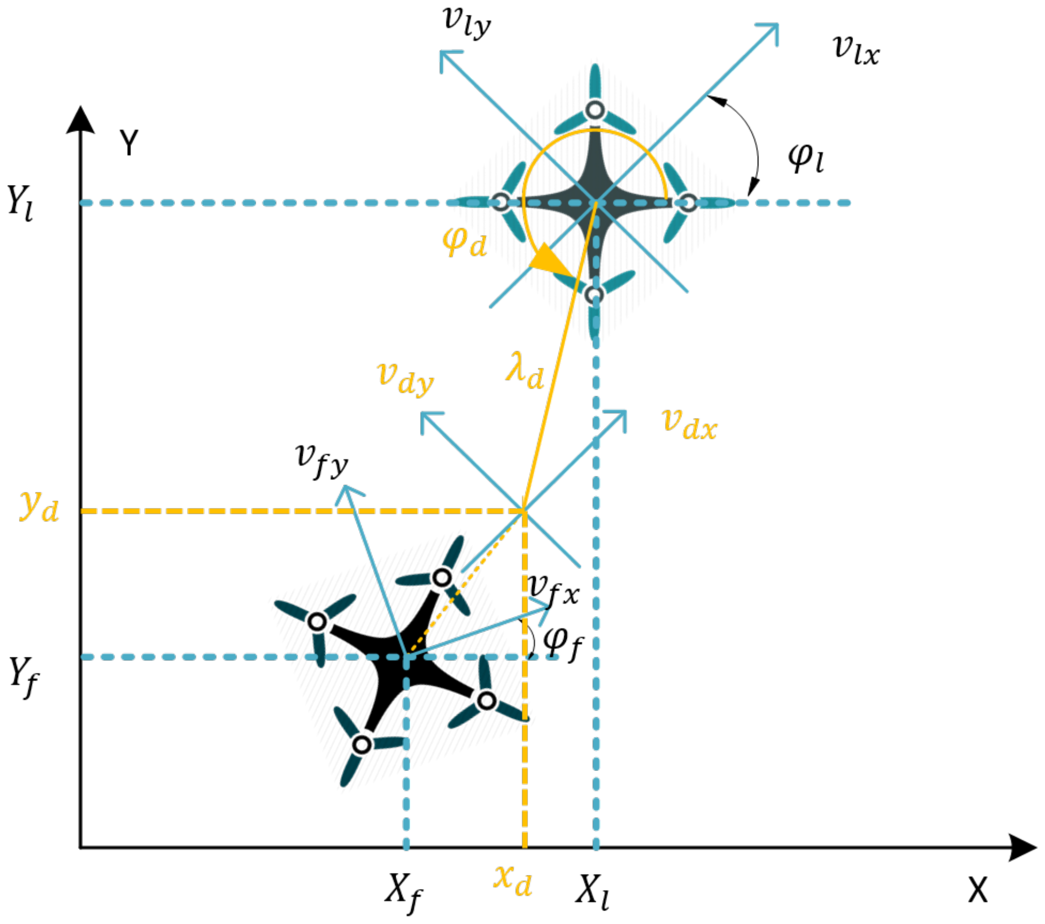


Figure 6.2. Illustration of the locations of leader and follower UAVs.

duce the controller design for the leader-follower tracking system. In particular, we introduce the formation and control algorithm, which include the position, yaw, and altitude controls.

6.3.1 Problem Formation

The formation of the leader and follower UAVs in the XY plane is shown in Figure 6.2. (X_i, Y_i) is the UAV's location. (v_{ix}, v_{iy}) is the velocities of the UAV. φ_i and w_i are the yaw and yaw rate of the UAV. i can be l (leader UAV), f (follower

UAV) and d (virtual desired follower UAV). (x_d, y_d) is the desired location from the leader UAV location. x_d and y_d are calculated as

$$\begin{aligned} x_d &= X_l + \lambda_d \cos(\varphi_d) \\ y_d &= Y_l + \lambda_d \sin(\varphi_d) \end{aligned} \tag{6.1}$$

where φ_d and λ_d are the desired heading and the desired distance with respect to the leader UAV in the XY coordinate. (x_d, y_d) is a virtual point where the follower UAV has the same velocity and yaw with the leader UAV. The state space functions of the UAVs can be described as

$$\begin{aligned} \dot{X}_i &= v_{ix} cos(\varphi_i) - v_{iy} sin(\varphi_i) \\ \dot{Y}_i &= v_{ix} sin(\varphi_i) + v_{iy} sin(\varphi_i) \\ \dot{\varphi}_i &= w_i \end{aligned} \tag{6.2}$$

The location of the follower UAV in the coordinate of the virtual follower UAV can be calculated as

$$\begin{aligned} \lambda_x &= -(x_d - X_f) cos(\varphi_d) - (y_d - Y_f) sin(\varphi_d) \\ \lambda_y &= (x_d - X_f) sin(\varphi_d) - (y_d - Y_f) cos(\varphi_d) \end{aligned} \tag{6.3}$$

Differentiating Equation (6.3) with respect to time and submitting Equation (6.2) into the result, we can obtain

$$\begin{aligned} \dot{\lambda_x} &= \lambda_y w_d + v_{fx} cos(e_\varphi) - v_{fy} sin(e_\varphi) - v_{dx} \\ \dot{\lambda_y} &= -\lambda_x w_d + v_{fx} sin(e_\varphi) + v_{fy} cos(e_\varphi) - v_{dy} \end{aligned} \tag{6.4}$$

where $e_\varphi = \varphi_f - \varphi_d$. Since we would like to design a sliding mode control system [102] for the follower UAV to arrive at the desired location (x_d, y_d) in the XY plane, the desired distances λ_x^d and λ_y^d are equal to 0. The tracking errors are defined as

$$\begin{aligned} e_x &= \lambda_x^d - \lambda_x = -\lambda_x \\ e_y &= \lambda_y^d - \lambda_y = -\lambda_y \\ e_\varphi &= \varphi_f - \varphi_d \end{aligned} \tag{6.5}$$

Differentiating of Equation (6.5) with time and using Equation (6.4), we have

$$\begin{aligned} \dot{e_x} &= e_y w_d - v_{fx} cos(e_\varphi) + v_{fy} sin(e_\varphi) + v_{dx} \\ \dot{e_y} &= -e_x w_d - v_{fx} sin(e_\varphi) - v_{fy} cos(e_\varphi) + v_{dy} \\ \dot{e_\varphi} &= w_f - w_d \end{aligned} \tag{6.6}$$

Equation (6.6) can be rewritten as

$$\dot{e} = F + Gu \tag{6.7}$$

where $\dot{e} = [e_x, e_y, e_\varphi]^T$, input $u = [v_{fx}, v_{fy}, w_f]^T$,

$$F = \begin{bmatrix} e_y w_d + v_{dx} \\ -e_x w_d + v_{dy} \\ -w_d \end{bmatrix} \tag{6.8}$$

$$G = \begin{bmatrix} -cos(e_\varphi) & sin(e_\varphi) & 0 \\ -sin(e_\varphi) & -cos(e_\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.9}$$

6.3.2 Position and Yaw Control Design

We define a sliding surface as

$$s = e + K_f \int e dt \tag{6.10}$$

where K_f is a constant matrix. When $\dot{s} = 0$, we can obtain

$$\dot{s} = \dot{e} + K_f e = F + Gu + K_f e = 0 \tag{6.11}$$

So that the position and yaw control input can be designed as

$$u = G^{-1}(-F - K_f e - L\,\mathrm{Sgn}(s)) \tag{6.12}$$

where L is a positive control parameter. Now we need to decide the range of L. We consider the measured data from sensors that have uncertainty errors δF so that Equation (6.7) can be rewritten as

$$\dot{e} = F + \delta F + Gu \tag{6.13}$$

where $||\delta F|| < L_1$, L_1 is a position constant. Consider the Lyapunov's function candidate as

$$V = 1/2s^T s \tag{6.14}$$

Differentiating (6.14) with respect to time, we have

$$\begin{aligned} \dot{V} &= s^T \dot{s} \\ &= s^T(F + \delta F + K_f e + Gu) \\ &= s^T(\delta F - L\,\mathrm{Sgn}(s)) \\ &\leq L_1||s|| - L||s|| \end{aligned} \tag{6.15}$$

When $\dot{V} <= 0$, we can obtain $L > L_1$.

6.3.3 Altitude Control Design

For the leader-follower tracking system, the desired altitude z_d of the follower UAV is the same as the altitude Z_l of the leader UAV. The altitude control is designed as

$$v_{fz} = k_{pz}(z_d - Z_f) \tag{6.16}$$

where v_{fz} is the velocity of the follower UAV at Z axis, k_{pz} is the proportional gain, and Z_f is the altitude of the follower UAV. Hence, the control input of the leader-follower tracking system is $(v_{fx}, v_{fy}, v_{fz}, w_f)$

6.4 Simulation Studies

In this section, the leader-follower tracking system is verified using Matlab and the HITL of Pixhawk with its control software PX4(V1.9.0) in the ROS Gazebo simulator.

6.4.1 Simulation in Matlab

In this simulation, we assume $\lambda_d = 20$, $\varphi_d = -2.618$, $L = 0.2$, $K_f = 0.2$, and $k_{pz} = 15$. The simulation time is $100s$ with sampling time of $0.1s$. To simulate the movement of the leader UAV, we use the smooth-turn mobility model [24] to generate the trajectory of the leader UAV in the XY plane. Meanwhile, the trajectory of the leader UAV is generated using the function $Z_l = 15 + 3sin(t/50)$, where t is time. The trajectories of the leader, follower and desired follower UAVs in XYZ coordinate and in XY plane are shown in Figure 6.3 and Figure 6.4, respectively. The follower UAV tracks the leader UAV at the desired positions. The relative distance between the follower UAV and the desired position at X axis, Y axis, and Z axis are shown in Figure 6.5, 6.6, and 6.7, respectively. Figure 6.8 shows the heading differences between the follower and leader UAVs.

6.4.2 Simulation in Gazebo

After setting up the simulation environment of PX4 in the ROS introduced in Chapter 4, we first generate multiple UAVs in the ROS Gazebo simulator using the command 'roslaunch px4 multi_uav_mavros_sitl.launch' in the PX4 firmware folder. Second, we launch the PX4 MAVROS node to convert the MAVLine messages of simulated PX4s to ROS topics using the commands 'roslanch MAVROS px4.launch fcu_rul:=udp://:14540@127.0.0.1:14557'. Third, we open the ground station Qgroundcontrol of PX4 to simulate the UAVs' movements on a Google map

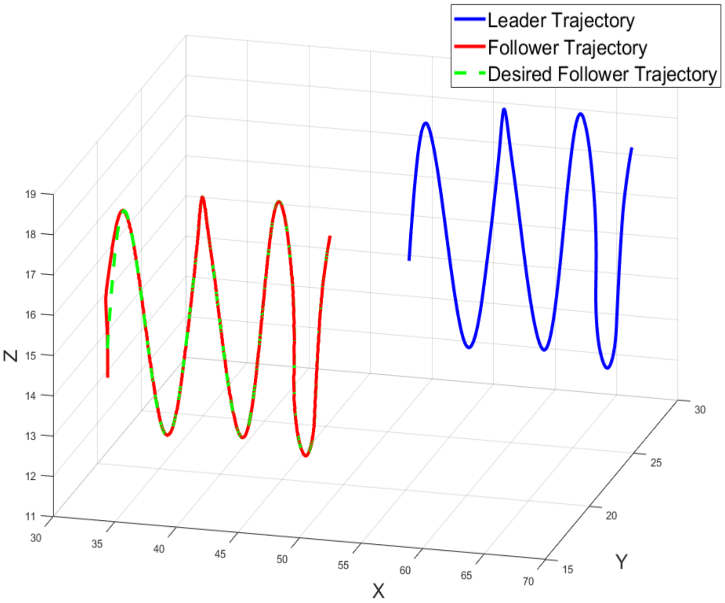


Figure 6.3. Trajectories of the UAVs in the XYZ coordinate.

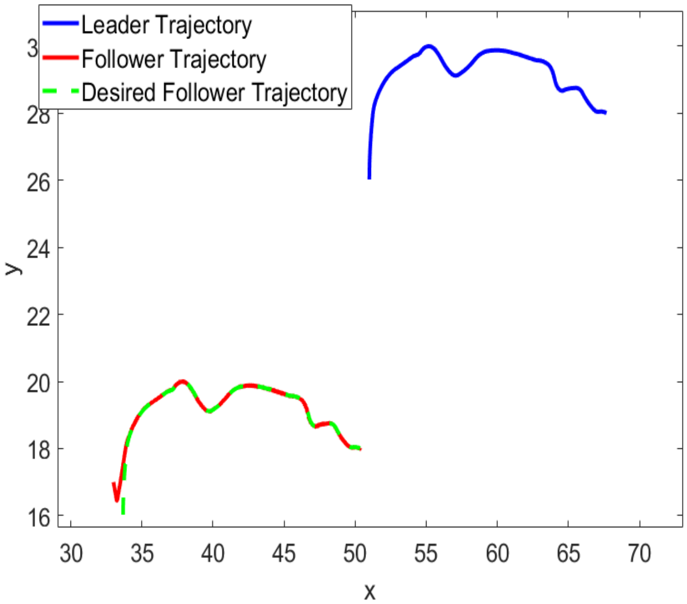


Figure 6.4. Trajectories of the UAVs in the XY plane.

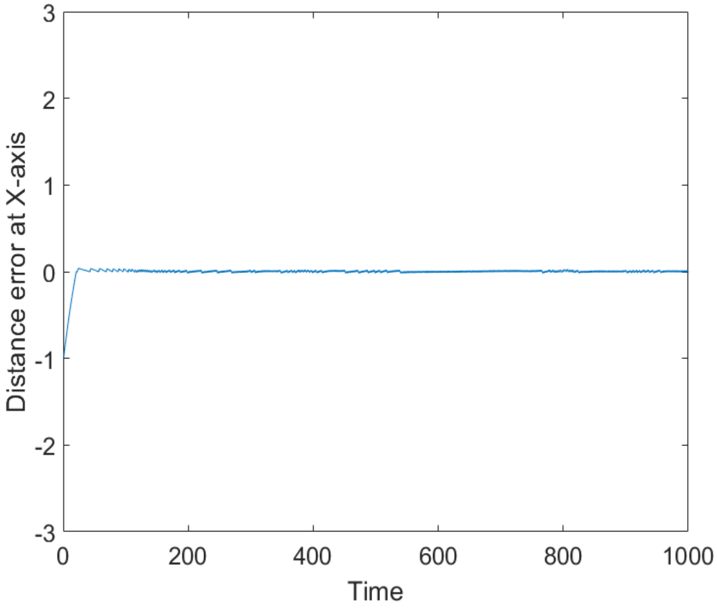


Figure 6.5. Relative distance between the follower UAV and desired position at X axis.

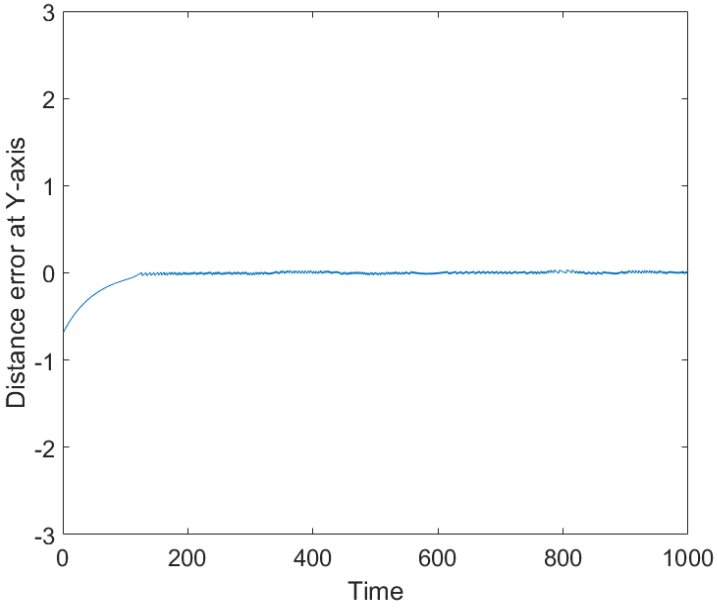


Figure 6.6. Relative distance between the follower UAV and desired position at Y axis.

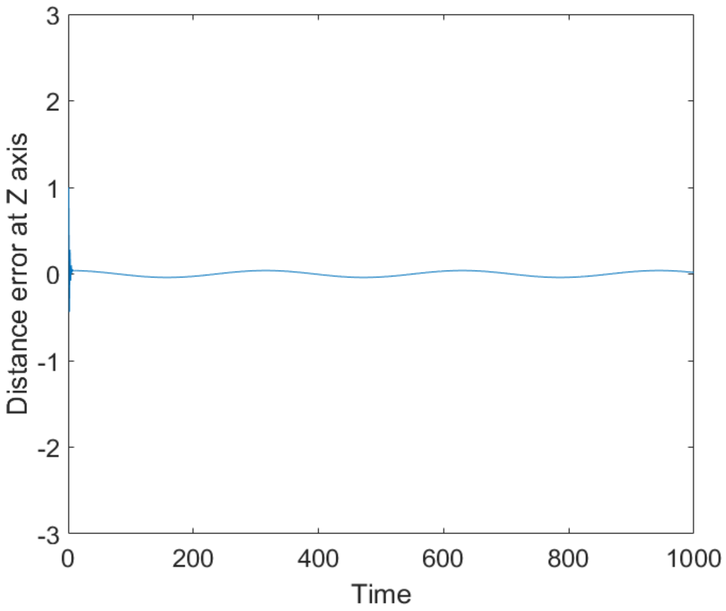


Figure 6.7. Relative distance between the follower UAV and desired position at Z axis.

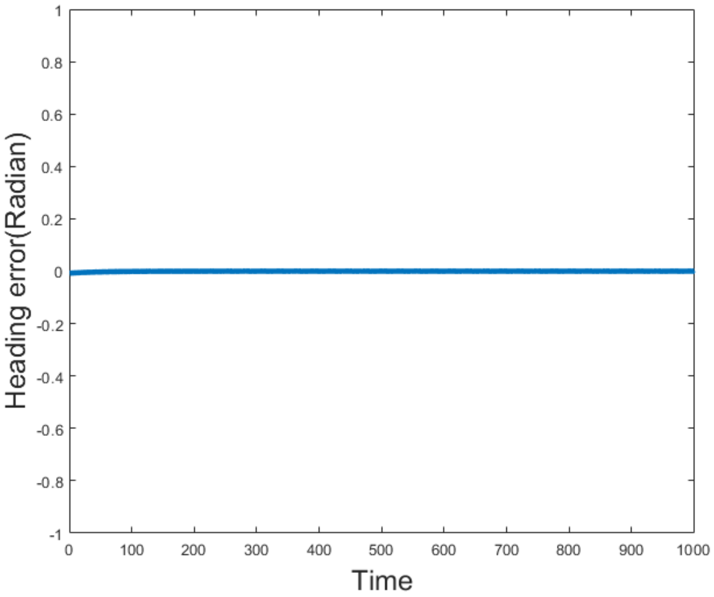


Figure 6.8. The yaw error between the follower UAV and desired position.

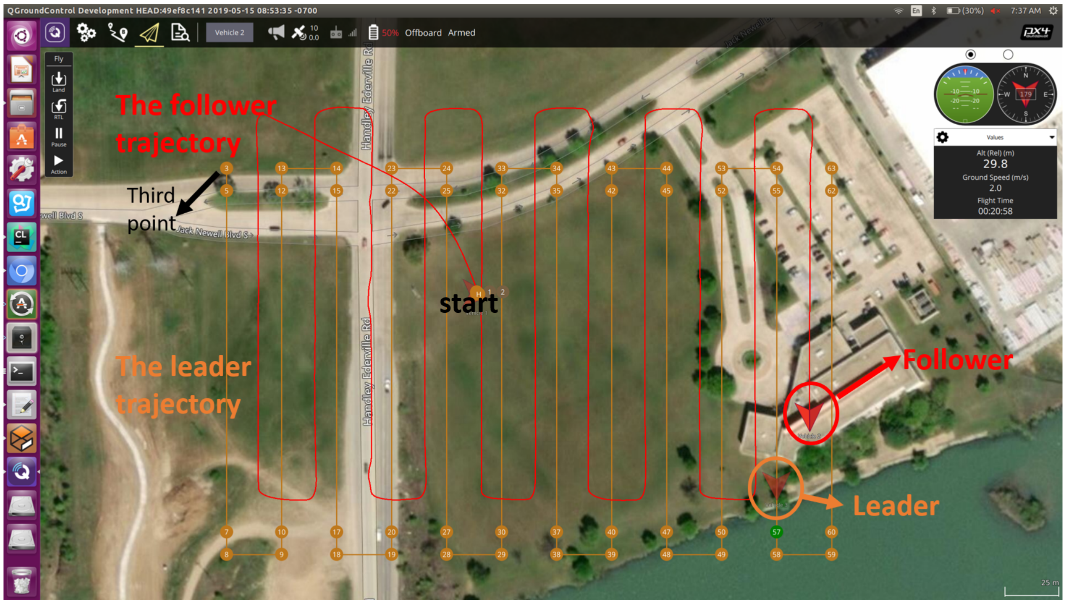


Figure 6.9. Simulation of leader-follower tracking system in ROS Gazebo.

with faked GPS sensors. Then, we plan a scan trajectory for the leader UAV. Finally, the leader-follower tracking controller is executed on the follower UAV, which subscribes to the leader UAV's ROS topics. The follower Pixhawk is controlled using '/remote/setpoint_raw/local' to control its velocities. The simulation result is shown in Figure 6.9.

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

This dissertation presented improvements made to the ACDA system, in terms of theory, implmentation, and applications. First, we developed a RL-based online directional antenna control solution for the ACDA system to establish a robust long-distance air-to-air communication channel using pure directional antennas. In particular, to capture the uncertain intentions of UAVs executing surveillance-like missions for better tracking, we adopted a UAV ST RMM. With this nonlinear random switching mobility model, a new state estimation algorithm that integrates MPCM and UKF was developed. To account for an unstable GPS environment and provide online optimal control solution, we developed a novel stochastic optimal controller by integrating RL and MPCM, which features a learning of communication RSSI models. With the learned RSSI model, the optimal solutions in both GPS-available and GPS-denied environments were developed. The learning and uncertainty-exploited decision framework is generally applicable to distributed decision-making of nonlinear multi-agent systems that are governed by random intentions in an uncertain environment.

Second, a new long-range broadband ACDA system which seamlessly integrates the communication, control and computing components was designed, implemented and tested. The communication component features a directional-antenna equipped UAV-to-UAV channel that shares the transmissions of the application data, control and commands, and the autonomous directional antenna control that focuses the energy and significantly extends the communication distance with reduced interference.

The computing and control components feature the use of RSSI, a communication performance indicator to assist with the antenna control to achieve optimal communication performance, and a RL-based control algorithm that learns the communication model in an unknown environment. We adopted ROS and BBB for the implementation of ACDA, and considered practical issues such as channel selection, interface design, application-oriented sensors and waster-resistant design for the practical use of the ACDA system in emergency response operations. The simulation studies, field tests, and disaster drills verified the performance of the system. The ACDA system can be broadly used to provide on-demand broad-band and long-distance UAV-to-UAV communication and service to the ground in places where communication infrastructures do not exit.

Third, we redesigned the ACDA system to improve the throughput and endurance. We also implemented BVLOS control of the remote UAV using ACDA. In particular, We changed the UAV platform to reduce interference, upgraded the computing component of ACDA with TX2 to enhance computing capability, designed a battery solution to extend flight time, and improved the rotation structure of the directional antenna to enhance the mechanical endurance of the tuning plate. We also connected the flight controller Pixhawk in the ACDA system to control the remote UAV via MAVROS. We implemented the BVLOS UAV control using a local laptop and verified it using simulation and initial sets of field test studies.

Then, we designed and implemented a remote UAV-based Mobile Health Monitoring System that supports long-distance communication and continuous tracking and monitoring. Our studies suggest that the use of a single camera and a single ball-shaped target fixed on the mobile infrastructure can find the relative position of the mobile target in the UAV's inertial frame. Based on the identified relative position, controllers are designed for the UAV's attitude and gimbal's pan and tilt rates for precise tracking.

Finally, we implemented a leader-follower tracking system based on the ACDA system. We first upgraded the ACDA system. Two UAV flight controller modules were connected the ACDA system. Then, we designed a control system to implement the leader-follower tracking.

7.2 Future Works

In the future work, we will extend the ACDA system to multi-UAV communication, by using phase-array-based electronically controlled directional antennas that have multiple controllable narrow beams. We will also build upon the ACDA system with distributed computing capabilities. In addition, we will also work with application domain experts on testing the system prototypes for practical uses.

REFERENCES

[1] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2624–2661, Fourthquarter 2016.

[2] H. Shakhatreh, A. H. Sawalmeh, A. I. Al-Fuqaha, Z. Dou, E. K. Almaita, I. M. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019.

[3] A. Shukla, H. Xiaoqian, and H. Karki, "Autonomous tracking and navigation controller for an unmanned aerial vehicle based on visual data for inspection of oil and gas pipelines," in *Proceedings of 2016 16th International Conference on Control, Automation and Systems (ICCAS)*, October 2016, pp. 194–200.

[4] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, "A uav system for inspection of industrial facilities," in *Proceedings of 2013 IEEE Aerospace Conference*, March 2013, pp. 1–8.

[5] P. Rudol and P. Doherty, "Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery," in *Proceedings of 2008 IEEE Aerospace Conference*, March 2008, pp. 1–8.

[6] J. Sun, B. Li, Y. Jiang, and C.-y. Wen, "A camera-based target detection and positioning uav system for search and rescue (sar) purposes," *Sensors*, vol. 16, p. 1778, October 2016.

[7] K. Ogura, Y. Yamada, S. Kajita, H. Yamaguchi, T. Higashino, and M. Takai, "Ground object recognition from aerial image-based 3d point cloud," in *Pro-*

ceddings of 2018 Eleventh International Conference on Mobile Computing and Ubiquitous Network (ICMU), October 2018, pp. 1–8.

[8] D. W. Casbeer, R. W. Beard, T. W. McLain, Sai-Ming Li, and R. K. Mehra, "Forest fire monitoring with multiple small uavs," in *Proceedings of the 2005, American Control Conference, 2005.*, June 2005, pp. 3530–3535 vol. 5.

[9] J. Han, Y. Xu, L. Di, and Y. Chen, "Low-cost multi-uav technologies for contour mapping of nuclear radiation field," *Journal of Intelligent and Robotic Systems*, vol. 70, no. 1, pp. 401–410, April 2013.

[10] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav-based iot platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, pp. 128–134, 2017.

[11] Y. Gu, M. Zhou, S. Fu, and Y. Wan, "Airborne wifi networks through directional antennae: An experimental study," in *Proceedings of 2015 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2015, pp. 1314–1319.

[12] J. Xie, F. Al-Emrani, Y. Gu, Y. Wan, and S. Fu, "Uav-carried long distance wi-fi communication infrastructure," in *Proceedings of AIAA Science and Technology Forum and Exposition*, January 2016.

[13] J. Chen, J. Xie, Y. Gu, S. Li, S. Fu, Y. Wan, and K. Lu, "Long-range and broadband aerial communication using directional antennas (acda): design and implementation," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 10 793–10 805, December 2017.

[14] J. Yan, Y. Wan, S. Fu, J. Xie, and S. Li, "Rssi-based decentralized control for robust long-distance aerial networks using directional antennas," *IET Control Theorey and Applications*, vol. 11, pp. 1838–1847(9), July 2017.

[15] S. Li, Y. Wan, S. Fu, M. Liu, and H. F. Wu, "Design and implementation of a remote uav-based mobile health monitoring system," in *Proceedings of SPIE on Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure*, vol. 10169, 2017. [Online]. Available: https://doi.org/10.1117/12.2257550

[16] B. Wang, J. Xie, S. Li, Y. Wan, S. Fu, and K. Lu, "Enabling high-performance onboard computing with virtualization for unmanned aerial systems," in *Proceedings of 2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2018, pp. 202–211.

[17] M. Liu, Y. Wan, S. Li, and F. L. Lewis, "Learning and uncertainty-exploited directional antenna control for robust aerial networking," in *Proceedings of 2019 IEEE 90th Vehicular Technology Conference (accepted)*, Honolulu, Hawaii, Sep 2019.

[18] M. Liu, Y. Wan, S. Li, F. Lewis, and S. Fu, "Learning and uncertainty-exploited directional antenna control for robust long-distance and broad-band aerial communication," *IEEE Transactions on Vehicular Technology (submitted)*, 2019.

[19] S. Li, Y. Gu, B. Subedi, C. He, and Y. Wan, "Beyond visual line of sight uav control for remote monitoring using directional antennas," in *Proceedings of IEEE GLOBECOM 2019 Workshop on Computing-Centric Drone Networks (submitted)*, Dec 2019.

[20] J. Yan, Y. Wan, J. Xie, S. Fu, and S. Li, "Rssi-based heading control for robust long-range aerial networking using directional antennas," in *Proceedings of 2017 American Control Conference (ACC)*, May 2017, pp. 5532–5537.

[21] S. Li, C. He, M. Liu, Y. Wan, Y. Gu, J. Xie, S. Fu, and K. Lu, "Design and implementation of aerial communication using directional antennas: learning control in unknown communication environments," *IET*

Control Theory Applications, April 2019. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-cta.2018.6252

[22] B. Wang, J. Xie, S. Li, Y. Wan, Y. Gu, S. Fu, and K. Lu, "Computing in the air: An open airborne computing platform," *IET Communications: UAV-Enabled Mobile Edge Computing 2019 (submitted)*, Nov 2019.

[23] Y. Wan and S. Fu, "Communicating in remote areas or disaster situations using unmanned aerial vehicles," *Homeland Security Today Magazine*, vol. 6, no. 7, pp. 32–35, 2015.

[24] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, "A smooth-turn mobility model for airborne networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3359–3370, Sep 2013.

[25] J. Xie, Y. Wan, J. H. Kim, S. Fu, and K. Namuduri, "A survey and analysis of mobility models for airborne networks," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 3, pp. 1221–1238, Third 2014.

[26] J. Xie, Y. Wan, K. Namuduri, S. Fu, G. L. Peterson, and J. F. Raquet, "Estimation and validation of the 3d smooth-turn mobility model for airborne networks," in *Proceedings of MILCOM 2013 - 2013 IEEE Military Communications Conference*, Nov 2013, pp. 556–561.

[27] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, vol. 9, pp. 32–50, Third 2009.

[28] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, Dec 2012.

[29] J. Xie, Y. Wan, K. Mills, J. J. Filliben, and F. L. Lewis, "A scalable sampling method to high-dimensional uncertainties for optimal and reinforcement learning-based controls," *IEEE control systems letters*, vol. 1, no. 1, pp. 98–103, July 2017.

[30] J. Xie, Y. Wan, and F. L. Lewis, "Strategic air traffic flow management under uncertainties using scalable sampling-based dynamic programming and q-learning approaches," in *Proceedings of 2017 11th Asian Control Conference (ASCC)*, Dec 2017, pp. 1116–1121.

[31] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes.* Tata McGraw-Hill Education, 2002.

[32] T. Li, Y. Wan, M. Liu, and F. L. Lewis, "Estimation of random mobility models using the expectation-maximization method," in *Proceedings of 2018 IEEE 14th International Conference on Control and Automation (ICCA)*, June 2018, pp. 641–646.

[33] T. S. Rappaport *et al.*, *Wireless communications: principles and practice.* PTR New Jersey, 1996, vol. 2.

[34] L. Josefsson and P. Persson, *Conformal array antenna theory and design.* John Wiley & Sons, 2006.

[35] Ubiquity, "Nanostation loco m5," *Ubiquiti Networks.* [Online]. Available: https://www.ubnt.com/airmax/nanostationm/

[36] J. D. Kraus, *Antennas.* McGraw-Hill Education, 1988.

[37] Y. Zhou, Y. Wan, S. Roy, C. Taylor, C. Wanke, D. Ramamurthy, and J. Xie, "Multivariate probabilistic collocation method for effective uncertainty evaluation with application to air traffic flow management," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 10, pp. 1347–1363, Oct 2014.

[38] S. Roy, D. Ramamurthy, and B. Lesieutre, “Studies on the probabilistic collocation method and its application to power system analysis,” in *Proceedings of 36th North America Power Symposium.* Moscow, ID, 2004.

[39] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.

[40] Y. Wan, S. Fu, J. Zander, and P. J. Mosterman, “Transforming on-demand communications with drones: The needs, analyses, and solutions,” *Homeland Security Today Magazine*, pp. 32–35, June 2015.

[41] S. Fu and Y. Wan, “Communicating in remote areas or disaster situations using unmanned aerial vehicles,” *HDIAC Journal*, vol. 2, pp. 4–8, 2016.

[42] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, “Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints,” *International Journal of Advanced Robotic Systems*, vol. 10, p. 28, 2013.

[43] G. Pajares, “Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs),” *Photogrammetric Engineering and Remote Sensing*, vol. 81, pp. 281–330, Jan 2015.

[44] C. Dixon and E. W. Frew, “Optimizing cascaded chains of unmanned aircraft acting as communication relays,” *IEEE Journal on Selected Areas in Communications*, vol. 30, pp. 883–898, June 2012.

[45] X. Dong, Y. Hua, Y. Zhou, Z. Ren, and Y. Zhong, “Theory and experiment on formation-containment control of multiple multirotor unmanned aerial vehicle systems,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 229–240, Jan 2019.

[46] S. Chandrasekharan, K. Gomez, A. Al-Hourani, S. Kandeepan, T. Rasheed, L. Goratti, L. Reynaud, D. Grace, I. Bucaille, T. Wirth, *et al.*, “Designing and

implementing future aerial communication networks," *IEEE Communications Magazine*, vol. 54, pp. 26–34, May 2016.

[47] C. Dixon and E. W. Frew, "Maintaining optimal communication chains in robotic sensor networks using mobility control," *Mobile Networks and Applications*, vol. 14, no. 3, pp. 281–291, Jun 2009. [Online]. Available: https://doi.org/10.1007/s11036-008-0102-0

[48] Z. Wang and G. B. Giannakis, "A simple and general parameterization quantifying performance in fading channels," *IEEE Transactions on Communications*, vol. 51, pp. 1389–1398, Aug 2003.

[49] K.-K. Wong, R. D. Murch, and K. B. Letaief, "Performance enhancement of multiuser mimo wireless communication systems," *IEEE transactions on communications*, vol. 50, pp. 1960–1970, Dec 2002.

[50] B.-C. Min, J. Lewis, E. T. Matson, and A. H. Smith, "Heuristic optimization techniques for self-orientation of directional antennas in long-distance point-to-point broadband networks," *Ad Hoc Networks*, vol. 11, pp. 2252–2263, 2013.

[51] B. Min, J. Lewis, D. K. Schrader, E. T. Matson, and A. Smith, "Self-orientation of directional antennas, assisted by mobile robots, for receiving the best wireless signal strength," in *Proceedings of 2012 IEEE Sensors Applications Symposium*, Feb 2012, pp. 1–6.

[52] TI, "Beaglebone black," *BeagleBoard.org*. [Online]. Available: https://beagleboard.org/black

[53] QT, "Software development framework," *QT — Cross-platform Software Development Framework*. [Online]. Available: https://www.qt.io/

[54] "Matrice 100," http://www.dji.com/matrice100/infospecs.

[55] Huawei, "Ws323 300mbps wireless range extender user guide," *Huawei Technologies*.

[56] ROS, "Robot operating system," *Ros.org.* [Online]. Available: http://www.ros.org/

[57] A. Gil, J. Segura, and N. M. Temme, *Numerical methods for special functions.* Siam, 2007.

[58] S. Dorafshan, R. J. Thomas, and M. Maguire, "Fatigue crack detection using unmanned aerial systems in fracture critical inspection of steel bridges," *Journal of Bridge Engineering*, vol. 23, no. 10, p. 04018078, 2018.

[59] J.-W. Kim, S.-B. Kim, J.-C. Park, and J.-W. Nam, "Development of crack detection system with unmanned aerial vehicles and digital image processing," in *Proceedings of 2015 World Congress on Advances in Structural Engineering and Mechanics (ASEM15)*, Incheon, Korea, August 2015.

[60] H. Yu, W. Yang, H. Zhang, and W. He, "A uav-based crack inspection system for concrete bridge monitoring," in *Proceedings of 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Fort Worth, TX, July 2017.

[61] H. Kim, J. Lee, E. Ahn, S. Cho, M. Shin, and S.-H. Sim, "Concrete crack identification using a uav incorporating hybrid image processing," *Sensors*, vol. 17, no. 9, p. 2052, 2017.

[62] M. Asnafi and S. Dastgheibifard, "A review on potential applications of unmanned aerial vehicle for construction industry," *Sustainable Structures and Materials, An International Journal*, vol. 1, no. 2, pp. 44–53, 2018.

[63] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "Uav-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, March 2017.

[64] NVIDIA, "Jetson tx2 module." [Online]. Available: https://developer.nvidia.com/embedded/jetson-tx2

[65] Tarot, "tarot 650." [Online]. Available: http://tarotrc.com/Product/Detail.aspx?Lang=enId=7641bc7f-e656-4d04-a075-13d7b81f96e0

[66] Pixhawk, "Pixhawk flight controller." [Online]. Available: https://docs.px4.io/en/flight_controller/mro_pixhawk.html

[67] Atmel, "Mega 2560." [Online]. Available: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf

[68] ROS, "rosserial," *Ros.org*. [Online]. Available: http://wiki.ros.org/rosserial

[69] ——, "Mavros," *Ros.org*. [Online]. Available: http://wiki.ros.org/mavros

[70] PX4, "Px4 user guide." [Online]. Available: https://docs.px4.io/en/

[71] Gazebo, "Gazebo simulator," *gazebosim.org*. [Online]. Available: http://gazebosim.org/

[72] Iperf. [Online]. Available: http://sourceforge.net/projects/iperf/

[73] C. Eschmann, C.-M. Kuo, C.-H. Kuo, and C. Boller, "Unmanned aircraft systems for remote building inspection and monitoring," in *Proceedings of the 6th European Workshop on Structural Health Monitoring, Dresden, Germany*, vol. 36, 2012.

[74] G. Morgenthal and N. Hallermann, "Quality assessment of unmanned aerial vehicle (uav) based visual inspection of structures," *Advances in Structural Engineering*, vol. 17, no. 3, pp. 289–302, 2014.

[75] Z. Yin, Y. Mao, and C. Seto, "Develop a uav platform for automated bridge inspection," *Mid-America Transportation Center University of Nebraska-Lincoln*, no. 25-1121-0003-295, March 2015.

[76] W. Jiang, F. Wenkai, and L. Qianru, "An integrated measure and location method based on airborne 2d laser scanning sensor for uav's power line inspection," in *Proceedings of 2013 Fifth International Conference on Measuring Technology and Mechatronics Automation*, Jan 2013, pp. 213–217.

[77] J. P. How, B. Behihke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE control systems*, vol. 28, no. 2, pp. 51–64, April 2008.

[78] B. Bethke, M. Valenti, and J. How, "Cooperative vision based estimation and tracking using multiple uavs," in *Advances in Cooperative Control and Optimization*, 2007, pp. 179–189.

[79] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, Nov 2013. [Online]. Available: https://doi.org/10.1007/s10514-013-9349-9

[80] V. N. Dobrokhodov, I. I. Kaminer, K. D. Jones, and R. Ghabcheloo, "Vision-based tracking and motion estimation for moving targets using small uavs," in *Proceedings of 2006 American Control Conference*, June 2006, pp. 6 pp.–.

[81] D. Deneault, D. Schinstock, and C. Lewis, "Tracking ground targets with measurements obtained from a single monocular camera mounted on an unmanned aerial vehicle," in *Proceedings of 2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 65–72.

[82] H. J. Asl and H. Bolandi, "Robust vision-based control of an underactuated flying robot tracking a moving target," *Transactions of the Institute of Measurement and Control*, vol. 36, no. 3, pp. 411–424, 2014.

[83] J. Pestana, J. L. Sanchez-Lopez, P. Campoy, and S. Saripalli, "Vision based gps-denied object tracking and following for unmanned aerial vehicles," in *Proceed-*

ings of 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Oct 2013, pp. 1–6.

[84] P. Zhan, K. Yu, and A. L. Swindlehurst, "Wireless relay communications with unmanned aerial vehicles: Performance and optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2068–2085, July 2011.

[85] Y. Zhou, N. Cheng, N. Lu, and X. S. Shen, "Multi-uav-aided networks: Aerial-ground cooperative vehicular networking architecture," *IEEE Vehicular Technology Magazine*, vol. 10, no. 4, pp. 36–44, Dec 2015.

[86] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1123–1152, Secondquarter 2016.

[87] J. Guan, F. Deboeverie, W. Philips, and et al, "Extrinsic calibration of camera networks using a sphere," *Sensors*, vol. 15, no. 8, pp. 18 985–19 005, 2015.

[88] "Raspberry pi," https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[89] "Pi camera," https://www.raspberrypi.org/documentation/hardware/camera/.

[90] "Ubiquity nanostation loco m5," https://www.ubnt.com/airmax/nanostationm/.

[91] X. Chen and Y. Jia, "Adaptive leader-follower formation control of non-holonomic mobile robots using active vision," *IET Control Theory Applications*, vol. 9, no. 8, pp. 1302–1311, 2015.

[92] D. Hartman, K. Landis, S. M. M. Mehrer, and J. Kim., "Quadcopter dynamic modeling and simulation," https://github.com/dch33/Quad-Sim.

[93] L. Garca-Carrillo, A. E. Dzul-Lpez, R. Lozano, and C. Pgard, *Quad Rotorcraft Control: Vision-Based Hovering and Navigation.* 1, pp. 32-34: Springer, 2013.

[94] H. Chao, M. Baumann, A. Jensen, Y. Chen, Y. Cao, W. Ren, and M. McKee, "Band-reconfigurable multi-uav-based cooperative remote sensing for real-time

water management and distributed irrigation control," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 11 744–11 749, 2008.

[95] A. Barrientos, J. Colorado, J. Cerro, A. Martnez-lvarez, C. Rossi, D. Sanz, and J. Valente, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *Journal of Field Robotics*, vol. 28, pp. 667–689, Aug 2011.

[96] M. Aljehani and M. Inoue, "Multi-uav tracking and scanning systems in m2m communication for disaster response," in *Proceedings of 2016 IEEE 5th Global Conference on Consumer Electronics*, Oct 2016, pp. 1–2.

[97] A. Sinha, A. Tsourdos, and B. White, "Multi uav coordination for tracking the dispersion of a contaminant cloud in an urban region," *European Journal of Control - EUR J CONTROL*, vol. 15, pp. 441–448, May 2009.

[98] P. Zarco-Tejada, V. Gonzalez-dugo, and J. A. J. Berni, "Fluorescence, temperature and narrow-band indices acquired from a uav platform for water stress detection using a micro-hyperspectral imager and a thermal camera," *Remote Sensing of Environment*, vol. 117, pp. 322–337, Feb 2012.

[99] Z. Li, Y. Liu, R. Hayward, J. Zhang, and J. Cai, "Knowledge-based power line detection for uav surveillance and inspection systems," in *Proceedings of 2008 23rd International Conference Image and Vision Computing New Zealand*, Nov 2008, pp. 1–6.

[100] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in *Proceedings of 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2, Sep. 2009, pp. 82–85.

[101] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-uav system," in *Proceedings of 2014 American Control Conference*, June 2014, pp. 2077–2082.

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

[103] R. Rafifandi, D. Lana Asri, E. Ekawati, and E. Mursito Budi, "Leaderfollower formation control of two quadrotor uavs," *SN Applied Sciences*, vol. 1, 06 2019.

[104] Z. Chao, L. Ming, Z. Shaolei, and Z. Wenguang, "Collision-free uav formation flight control based on nonlinear mpc," in *Proceedings of 2011 International Conference on Electronics, Communications and Control (ICECC)*, Sep. 2011, pp. 1951–1956.

[105] Y. Chen, J. Yu, X. Su, and G. Luo, "Path planning for multi-uav formation," *Journal of Intelligent Robotic Systems*, vol. 77, Jan 2015.

[106] X. Dong, G. Cai, F. Lin, B. M. Chen, H. Lin, and T. H. Lee, "Implementation of formation flight of multiple unmanned aerial vehicles," in *Proceedings of IEEE ICCA 2010*, June 2010, pp. 904–909.

[107] B. Yun, B. M. Chen, K. Y. Lum, and T. H. Lee, "Design and implementation of a leader-follower cooperative control system for unmanned helicopters," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 61–68, Feb 2010. [Online]. Available: https://doi.org/10.1007/s11768-010-9188-6

BIOGRAPHICAL STATEMENT

Songwei Li received the B.S. degree from Nanchang Hangkong University, Jiangxi Province, China in 2010, and the M.S. degree from University of North Texas, Denton, Texas in 2016. He is now working toward the Ph.D. degree in the department of Electrical Engineering at the University of Texas at Arlington, Arlington, Texas. His research interests include embedded control system, robotics, intelligent control systems in unmanned vehicles, and embedded wireless sensor networks.