Experimental Validation of a Distributed Control Approach Based on Multiplex Networks on Formations of Unmanned Aerial Vehicles

Jesse Jaramillo[*†], Tyler Wieczorek[*†], Dzung Tran[†],
Venkata Sireesha Dadi[‡], Tansel Yucelen[§], and Sriram Chellappan[¶]

Standard distributed control methods for vehicle formation assignments are restricted to utilizing single-layer of consensus or consensus-like algorithms, which result in multiagent systems with fixed spatial properties. To address this problem, the authors of this paper recently proposed a new distributed architecture for formation control in which multiple-layers of consensus or consensus-like algorithms (multiplex information networks) are utilized to enable the flexibility in alternating the size and orientation of the resulting formation without requiring global information exchange ability. In this paper, this new architecture is implemented to generate guidance commands for a group of unmanned aerial vehicles to address the formation tracking problem in a three-dimensional space. In particular, small-size quadcopter platforms are used for laboratory-level experiments and a low-cost local positioning system is utilized to obtain the absolute positions of the platforms in that space. The experimental results validate the feasibility and efficacy of the proposed architecture for practical problems.

I Introduction

Standard distributed control methods for vehicle formation assignments are restricted to utilizing single-layer of consensus or consensus-like algorithms, which result in multiagent systems with fixed spatial properties; that is, once formations are formed, their size and orientation cannot be changed. To address this problem, the authors of this paper recently proposed a new distributed architecture for formation control in Refs. 1–3 in which multiple-layers of consensus or consensus-like algorithms (multiplex information networks) are utilized to enable flexibility in alternating the size and orientation of the resulting formation without requiring global information exchange ability. To elucidate this point, Figure 1 illustrates how the formation's size and orientation are distributively controlled by the proposed architecture. Specifically, the architecture consists of a main layer for creating and maintaining the formation and multiple layers of consensus for distributing

[*]The authors contributed equally and are listed alphabetically.

[†]J. Jaramillo, T. Wieczorek, and D. Tran are Graduate Research Assistants of the Mechanical Engineering Department and Members of the Laboratory for Autonomy, Control, Information, and Systems (LACIS, `http:/http://lacis.eng.usf.edu/`) at the University of South Florida, Tampa, Florida 33620, United States of America (emails: `jessej1@mail.usf.edu, twieczorek@mail.usf.edu, dtran3@mail.usf.edu`).

[‡]V.S. Dadi is a Graduate Research Assistant of the Computer Science and Engineering Department at the University of South Florida, Tampa, Florida 33620, United States of America (email: `dadi@mail.usf.edu`).

[§]T. Yucelen is an Assistant Professor of the Mechanical Engineering Department and the Director of the Laboratory for Autonomy, Control, Information, and Systems (LACIS, `http://http://lacis.eng.usf.edu/`) at the University of South Florida, Tampa, Florida 33620, United States of America (email: `yucelen@usf.edu`).

[¶]S. Chellappan is an Associate Professor of the Computer Science and Engineering Department at the University of South Florida, Tampa, Florida 33620, United States of America (email: `sriramc@usf.edu`).

[‖]This work was supported in part by a grant from Florida High Tech Corridor. Any results, opinions and conclusions are those of the authors alone, and do not necessarily reflect the views of the funding agency.

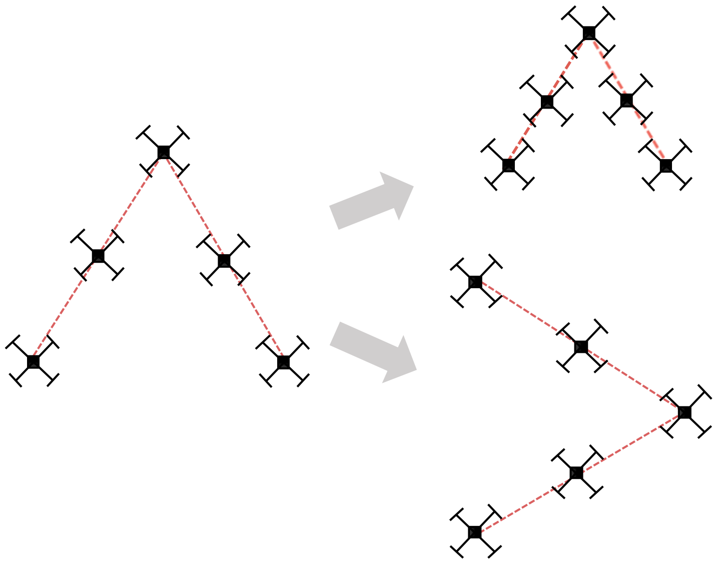

Figure 1. The formation size and orientation can be controlled via multiplex information network[1–3].

the information that is only available to the capable vehicles (or leaders) such as the desired scaling factors and the desired rotation angles. The updates in other layers affect the main layer simultaneously and lead to the change in the resulting formations.

While the proposed distributed architecture has been implemented on unmanned ground robots in Ref. 3, the implementation in a three-dimensional space for unmanned aerial vehicles is more challenging because of the increase in complexity. That is, our intention in this paper is to verify the feasibility and efficacy of the architecture proposed in Refs. 1–3 by implementing it to generate guidance commands for a group of unmanned aerial vehicles to address the formation tracking problem in the three-dimensional space. In particular, the Crazyflie nano-quadcopters in Ref. 4 are used as the platforms for laboratory-level experiments. Furthermore, the loco positioning system is utilized to obtain the absolute positions of the platforms in that space. In what follows, we use $\mathbb{R}$ for the set of real numbers, $\mathbb{R}^n$ for the set of $n \times 1$ real column vectors, $\mathbb{R}^{n\times m}$ for the set of $n \times m$ real matrices, $\mathbb{R}_+$ (respectively, $\overline{\mathbb{R}}_+$) for the set of positive (respectively, nonnegative) real numbers, $\mathbb{R}_+^{n\times n}$ (respectively, $\overline{\mathbb{R}}_+^{n\times n}$) for the set of $n \times n$ positive-definite (respectively, nonnegative-definite) real matrices, and $\mathrm{diag}(a)$ for the diagonal matrix with the vector a on its diagonal. In addition, we write $(\cdot)^{\mathrm{T}}$ for the transpose operator, $(\cdot)^{-1}$ for the inverse operator, $\det(\cdot)$ for the determinant operator, $\mathrm{tr}(\cdot)$ for the trace operator, and $\mathrm{sgn}(\cdot)$ for the signum function.

II Control Architecture and Multiplex Information Networks

We consider a two-level control hierarchy, which consists of a low-level control law for command tracking and a high-level guidance law for distributed formation control of the vehicles. Specifically, the low-level control law is a PID controller implemented on the platform vehicle to track a commanded position while the high-level guidance law is the distributed formation control architecture from Refs. 1–3, which utilizes multiplex information networks to generate the guidance commands for the low-level control law. Throughout this paper, we only focus on the architecture implemented at the high-level guidance law. For this purpose, we consider a group of N aerial vehicles exchanging information among each other using their local measurements according to an undirected and connected graph $\mathcal{G}$. Let $x_i(t) \in \mathbb{R}^3$, $i = 1, 2, \dots, N$ be the guidance command generated by the high-level distributed control law. The resulting guidance dynamics can be described by the single integrator state $x_i(t) \in \mathbb{R}^3$ satisfying the equation given by $\dot{x}_i(t) = u_i(t)$, $x_i(0) = x_{i0}$, $i = 1, 2, \dots, N$

with $u_i(t) \in \mathbb{R}^3$.

The proposed distributed formation control architecture utilizing multiplex information network is implemented at the high-level guidance law with a main layer for creating and maintaining the formation and multiple layers of consensus for distributing the desired parameters that is only available to the capable vehicles[1]. In particular, the dynamics of the main layer is given by

$$\begin{aligned} \dot{x}_i(t) &= -\sum_{j \in \mathcal{N}_i^f} \Big(\big(x_i(t) - p_i(t) - c_i(t)\big) - \big(x_j(t) - p_j(t) - c_j(t)\big) \Big) \\ &\quad -k_i\big(x_i(t) - p_i(t) - c_i(t)\big) + \dot{p}_i(t) + \dot{c}_i(t), \quad x_i(0) = x_{i0}, \end{aligned} \tag{1}$$

where $x_i(t) \in \mathbb{R}^3$ denotes the state (i.e., physical position) of vehicle i, $c_i(t) \triangleq [c_i^x(t), c_i^y(t), , c_i^z(t)]^{\mathrm{T}} \in \mathbb{R}^3$ is the tracking command or the dynamic target position, and

$$p_i(t) \triangleq R\big(\theta_i^z(t), \theta_i^y(t), \theta_i^x(t)\big) S\big(\gamma_i^x(t), \gamma_i^y(t), \gamma_i^z(t)\big)\xi_i, \tag{2}$$

correspond to the signals locally obtained through other network layers described below. In (1), $k_i = 1$ only for capable vehicles and $k_i = 0$ otherwise. We consider that there is at least one capable vehicle in the multiagent system. In (2), $\xi_i \in \mathbb{R}^3$ denotes the desired position of vehicle i in the reference formation, $\theta_i^x(t) \in \mathbb{R}$, $\theta_i^y(t) \in \mathbb{R}$, and $\theta_i^z(t) \in \mathbb{R}$ are the rotation angles corresponding to roll, pitch, and yaw respectively, $R\big(\theta_i^z(t), \theta_i^y(t), \theta_i^x(t)\big)$ is the rotation matrix, $\gamma_i^x(t) \in \mathbb{R}$, $\gamma_i^y(t) \in \mathbb{R}$, and $\gamma_i^z(t) \in \mathbb{R}$ are the scaling factors in x, y, and z axes, respectively, and $S\big(\gamma_i^x(t), \gamma_i^y(t), \gamma_i^z(t)\big) \triangleq \mathrm{diag}([\gamma_i^x(t), \gamma_i^y(t), \gamma_i^z(t)]^T)$ is the scaling matrix.

In order to define the dynamical structure of other network layers, let $\phi_i(t)$ denotes either $c_i^x(t) \in \mathbb{R}$, $c_i^y(t) \in \mathbb{R}$, $c_i^z(t) \in \mathbb{R}$, $\theta_i^x(t) \in \mathbb{R}$, $\theta_i^y(t) \in \mathbb{R}$, $\theta_i^z(t)$, $\gamma_i^x(t) \in \mathbb{R}$, $\gamma_i^y(t) \in \mathbb{R}$, or $\gamma_i^z(t) \in \mathbb{R}$ and has the following dynamics

$$\dot{\phi}_i(t) = -q_i(t) - \tau \mathrm{sgn}\big(q_i(t)\big), \ \phi_i(0) = \phi_{i0}, \tag{3}$$

$$q_i(t) \triangleq \sum_{j \in \mathcal{N}_i^f} \big(\phi_i(t) - \phi_j(t)\big) + k_i\big(\phi_i(t) - \phi_0(t)\big), \tag{4}$$

where $\tau \in \mathbb{R}$ is a positive design parameter and it is assumed that $\phi_0(t)$ and $\dot{\phi}_0(t)$ are bounded (i.e., $|\phi_0(t)| \leq \bar{\phi}_0$ and $|\dot{\phi}_0(t)| \leq \bar{\phi}$). The parameter τ is chosen such that $\tau > \bar{\phi}$. Note that since $k_i = 1$ only for capable vehicles in (3) and (4), $\phi_0(t)$ denotes the corresponding information that is available only to the capable vehicles such as the position of the dynamic target in the space $c(t) \triangleq [c^x(t), c^y(t), c^z(t)]^{\mathrm{T}}$, the desired rotation angle about the x-axis $\theta_0^x(t)$, the desired scaling factor in x-axis $\gamma_0^x(t)$, etc.

The proposed architecture given by (1), (2), (3), and (4) allows the vehicles not only to create and maintain the desired formation and track a dynamic target, but to also alter the size and orientation of the resulting formation. While the theoretical proofs of the above statement for solving the problem in planar plane can be found in Refs. 1-3, the proofs can be easily extended to three-dimensional space in the same manner as also noted in Ref. 3.

III Experiment Setup

In this section, we present the experimental setup and provide more information on the platforms used. Specifically, we first note that the proposed distributed architecture is implemented in a pseudo-distributed manner; that is, one workstation computer is deployed to calculate the distributed control signals (i.e., the proposed architecture) at the high-level guidance laws of all aerial platforms. The information utilized in generating the control signal for each platform is regulated to a level of communication with some certain neighboring platforms. The control signals, which play the role as guidance commands, are then appropriately forwarded to each platform. Once a platform receives the command, the low-level control law implemented at the firmware level executes and tracks the command accordingly.

[1]Capable vehicles denote vehicles that have the knowledge of desired parameters used to control the resulting formation.

A Crazyflie

The platform being used to perform the experiment is the Crazyflie 2.0 nano-quadcopter from Bitcraze in Ref. 4 owing to its open-source nature and robustness. Specifically, this quadcopter runs a MPU-6050 chip on board, which has a 3-axis gyroscope and 3-axis accelerometer. It weighs 27 grams with a maximum payload of 15 grams and uses a Lithium-Polymer battery with a flight time of approximately seven minutes, depending on the payload. In addition, the experiment requires the implementation of Bitcraze's loco positioning deck and flow deck at the top and bottom of the Crazyflie, respectively, as illustrated in Figure 2. The flow deck has two sensors. The first sensor is a laser range sensor allowing this quadcopter to measure the distance to the ground up to two meters. The second is an optical flow sensor, which allows the vehicle to monitor how the ground is moving in relation to itself for the purpose of preventing the drifting phenomenon. The loco positioning deck allows for communication with the loco positioning system, which is explained later in the description of the loco positioning system. Both decks are used in conjunction with the loco positioning system.

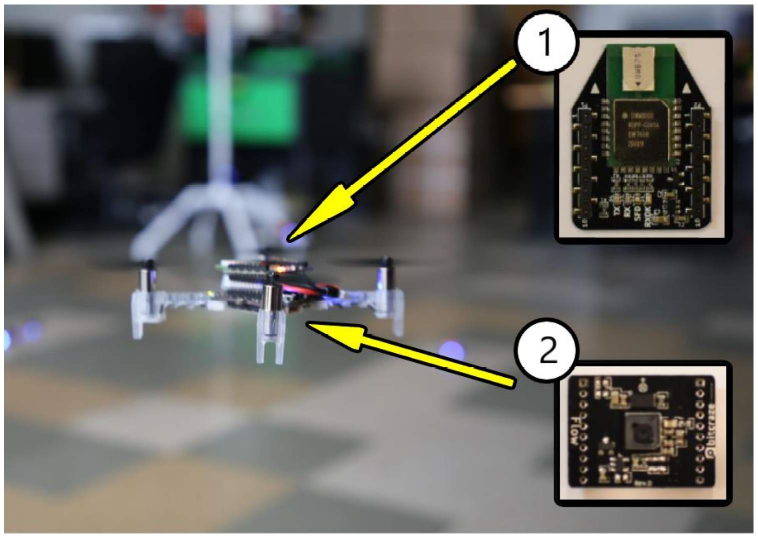

Figure 2. The Crazyflie nano-quadcopter[4] with 1) the loco positioning deck[5] on the top and 2) the flow deck[6] at the bottom.

The Crazyflie's firmware is written in Python 3.6 and the communication is necessary between a workstation computer and this quadcopter. Establishing a connection is done through the use of Bitcraze's Crazyradio, which is a 2.4 GHz USB radio dongle. It is attached to the workstation and transmits necessary control signals to as well as receives onboard sensors' data from the Crazyflie. Bitcraze provides a GUI PC client allowing users to set the Crazyflie's address, channel, and bandwidth. The GUI PC client is also used to configure the loco positioning system, which will be explained in the following subsection. For the experiment, the address and bandwidth are kept the same for each Crazyflie, but the channel is different. This is done to allow multiple signals to be sent and received without significant packet loss. The available bandwidths are 250 Kbit/s, 1 Mbit/s, and 2 Mbit/s. The bandwidth being used in the experiment is 2 Mbit/s, which allows for greater reliability when flying multiple Crazyflie quadcopters.

B Loco Positioning System

The loco positioning system (LPS) being used in this experiment consists of 8 positioning anchors and a loco positioning deck for each Crazyflie. Using the Crazyflie PC Client, the initial anchor positions are assigned and numbered from 0-7 with the anchors arranged in a $3m$ x $5m$ x $2m$ (width x length x height) rectangular prism as showed in Figure 3. The LPS has two different protocols that can be used to communicate with aerial vehicles: two way ranging (TWR) and time distance of arrival (TDoA). TWR protocol expects only one Crazyflie is being flown and thus does not work for swarm formations. Instead this experiment uses

TDoA to handle answering of ranging requests from each anchor and subsequent Crazyflie quadcopters within the system parameters aforementioned. The system uses TDoA protocol to form a global coordinate system in which each Crazyflie with a loco positioning deck receiving continuous pings sent by the anchors. Each quadcopter first perceives its position by the difference between receive time of the anchors, which is calculated as distance from those anchors. Next, it sends a packet of positioning information to the workstation computer through the radio dongle. The computer then sends updated positioning information via the dongle to each individual Crazyflie with respect to the LPS global positioning. In addition, each anchor has a time slot of 2ms to send its packet to the Crazyflie quadcopters (8 time slots for the 8 anchors equaling a unit frame length of 16ms). All anchors use the same channel to send information to the loco positioning decks implemented on the Crazyflie quadcopters. Furthermore, anchor number 0 is considered the master anchor that starts off the sequence of time slots for packet communication. The LPS has an accuracy of 0.1 meter in measuring the absolute position. Using this system with the Bitcraze flow deck onboard the Crazyflie minimizes potential drifting while in flight within a swarm formation.

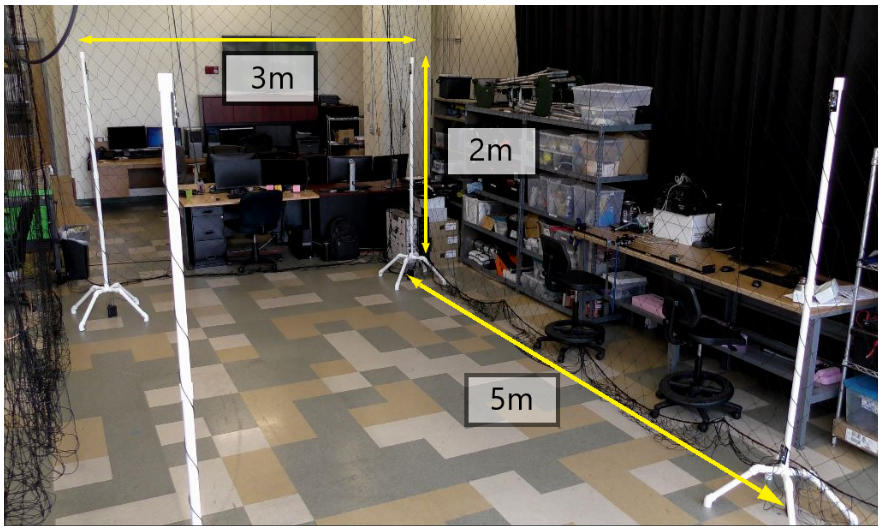

Figure 3. The loco positioning system (LPS) utilized in the laboratory-level experiments.

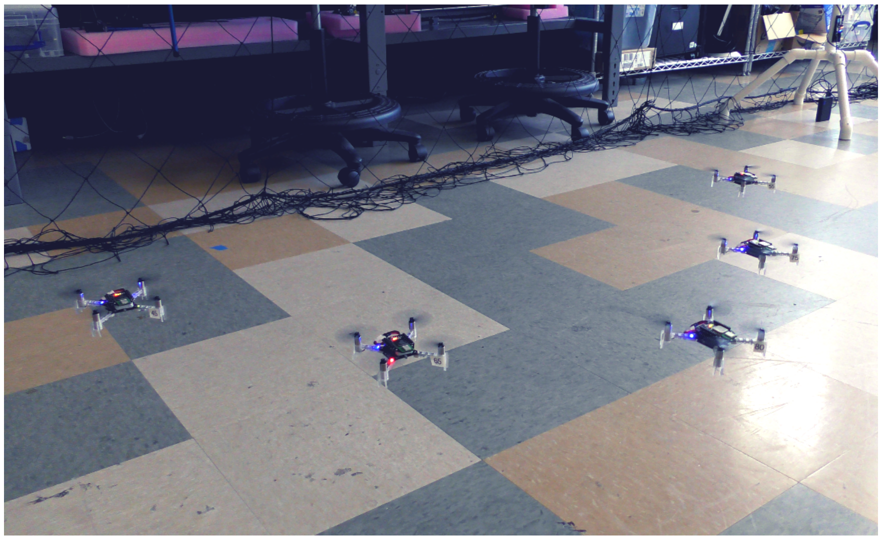

Figure 4. A case with 5 Crazyflie quadcopters in V-formation.

C Topology Setup

For the experiment, a group of 5 Crazyflies is utilized to form a desired V-formation (Figure 4) with the communication topology and desired reference formation depicted in Figure 5. Agent 3 is the leader and the other four are followers. The leader receives information for command following $c^x(t)$ and $c^y(t)$ in the x and y directions, scaling factors $\gamma_0^x(t)$ and $\gamma_0^y(t)$ in the x and y directions, and rotation angle $\theta_0^x(t)$ of the entire formation. Initially, the five agents are all positioned with their fronts facing in the positive x direction.

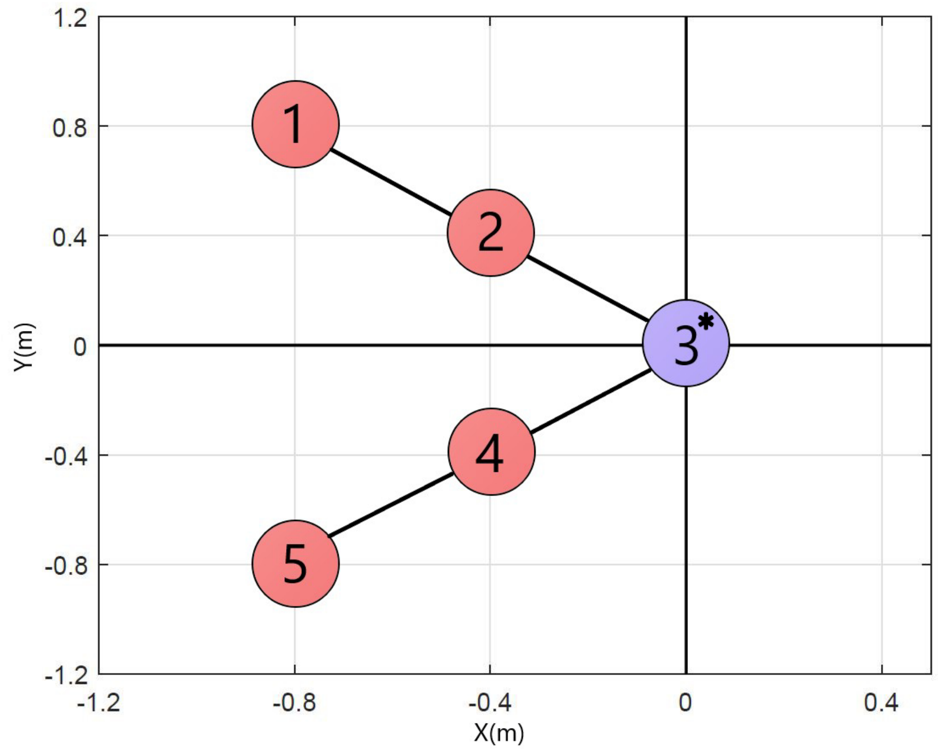

Figure 5. Order and communication neighborhood of each aerial vehicle.

IV Experimental Results

This section discusses the results of implementing the multiplex control architecture (1), (2), (3), and (4) for distributed formation control. The experiment is divided into three consecutive stages: Establishing, rotating, and scaling the formation. Specifically, Figure 6 presents the evolution of the group of quadcopters with the commands $\theta_0^x(t) = 0$ and $\gamma_0^x(t) = \gamma_0^y(t) = 1$ sent to the leader. Each vehicle starts moving from the initial position and adjust themselves to establish the reference formation at the end of the stage. Next, the desired rotation angle is changed to $\theta_0^x(t) = \pi/2$ and the desired scaling factors are still kept at $\gamma_0^x(t) = \gamma_0^y(t) = 1$. Figure 7 shows that the formation starts rotating 90° counterclockwise toward the desired configuration while the V-formation is maintained during the rotation. Finally, the leader vehicle (agent 3) is commanded to lead the V-formation through the obstacles as shown in Figure 8.

As the V-formation approaches the obstacles (shown as black squares), the commanded scaling factor $\gamma_0^y(t) = 0.4$ is sent to the leader to make the formation smaller so as to not hit the obstacles. As a response, the formation is contracted in the y direction of the reference formation (x direction of the LPS global frame) to pass through the obstacles. The scaling factor is then changed back to $\gamma_0^y(t) = 1$ to recover the initial formation. We also note that, in Figure 8 the vehicles on the farthest edges of the formation seems to experience some degrees of time delay, which is an ubiquitous phenomenon in multiagent system. An algorithm to compensate for time delay in formation control will be investigated as a future research. Overall, this experiment clearly validates the feasibility and efficacy of the multiplex control architecture for formation control.

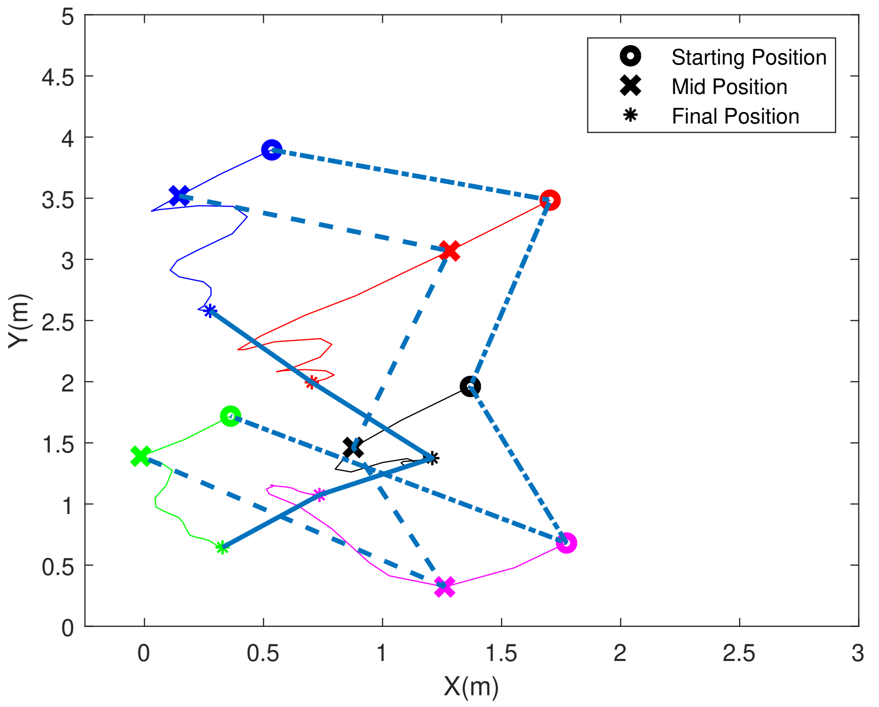

Figure 6. **5 Crazyflies forming a V-formation from arbitrary starting positions, where lines show the various points of the formation; dashed-dotted lines show the initial formation, dashed lines show formation in the middle of motion and the solid lines show the final position of the layer.**

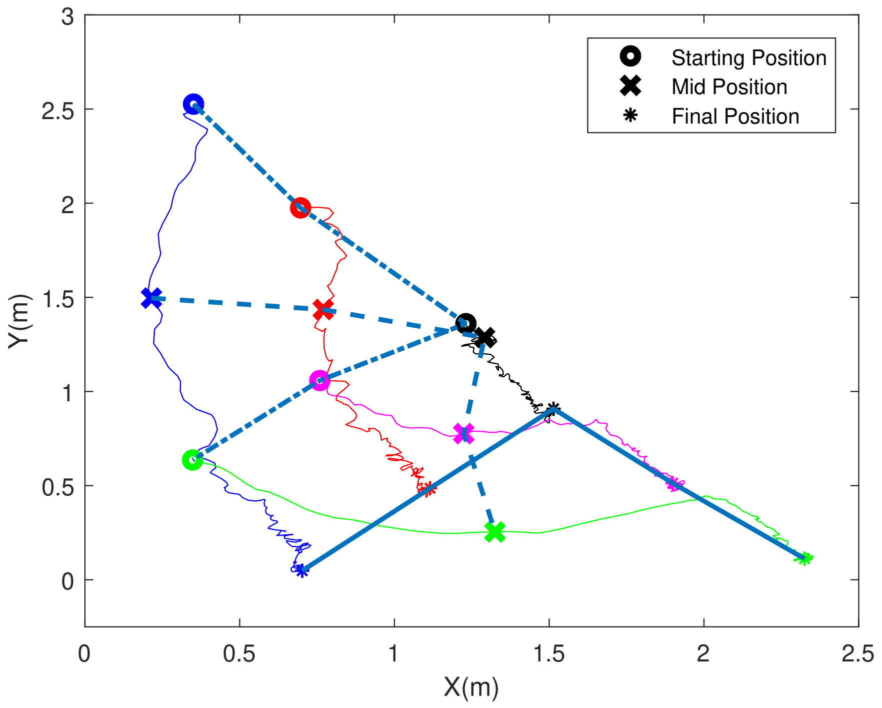

Figure 7. **The V-formation is rotating counterclockwise 90 degrees, where lines show the various points of the formation; dashed-dotted lines show the initial formation, dashed lines show formation in the middle of motion and the solid lines show the final position of the layer.**

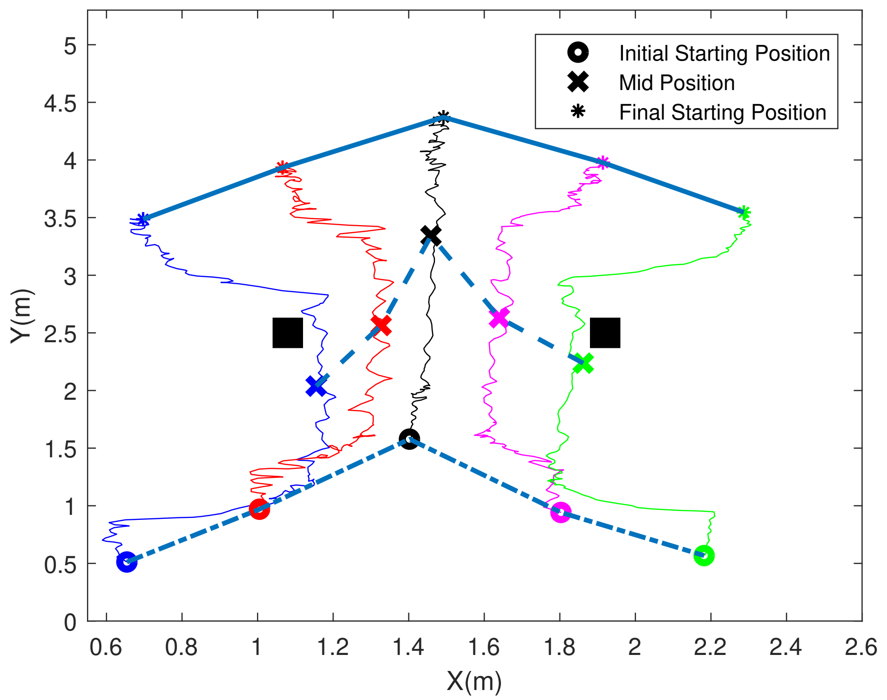

Figure 8. The V-formation contracts when reaching the obstacles and expands out afterwards, where lines show the various points of the formation; dashed-dotted lines show the initial formation, dashed lines show formation in the middle of motion and the solid lines show the final position of the layer.

We now present another experimental verification through a practical application that tests sending the scaling factors, $\gamma_0^x(t)$ and $\gamma_0^y(t)$, to the group of 5 Crazyflies via arm motion using a wearable Application Program Interface (API). The API being used is a first generation Moto 360 smartwatch shown in Figure 9 and in Ref. 7. In order to track the arm movements of the user, a gyroscope sensor within the smartwatch is used. The sensor provides three-dimensional position values (x, y and z axes). The gyroscope calculates the rotational vector and gives the values into the array with the indices ranging from 0 to 2. The values start with 0 being the index for x axis, 1 for the y axis, and 2 for the z axis, which tracks the left-right, forward-backward, and up-down motions, respectively. These values provide guidelines for the smartwatch to detect the motion and process them as signals. The smartwatch then sends this information to a smartphone via bluetooth. The smartphone then processes this information and sends it to the workstation via wifi connection using socket programming and port-to-port communication. The IP address and port number are obtained from the workstation and entered into the phone's application to establish a connection. The communication setup used in this experiment is shown below in Figure 10. If the watch detects a motion to the left, the desired scaling factors in x and y directions are reduced by 0.5 from the current ones and sent to the leader while a motion to the right adds 0.5 to the current scaling factors. Shown in Figure 11 is the V-formation of the 5 Crazyflies when given different scaling factors via the watch. Specifically, the reference formation used in this experiment is the same as in Figure 5 with the initial scaling factors set to $\gamma_0^x(t) = \gamma_0^y(t) = 1$. The formation is then commanded to shrink with an arm motion to the left (i.e., $\gamma_0^x(t) = \gamma_0^y(t) = 0.5$). Then, two consecutive arm motions to the right are made to adjust the scaling factors to $\gamma_0^x(t) = \gamma_0^y(t) = 1.5$. Under the multiplex control architecture, the formation responses accordingly as expected.

V Conclusion

In this paper, we experimentally validated the proposed multiplex networks-based distributed control architecture in a three-dimensional space using aerial vehicles. Specifically, command following and formation

Figure 9. Moto 360 1st Generation[7].

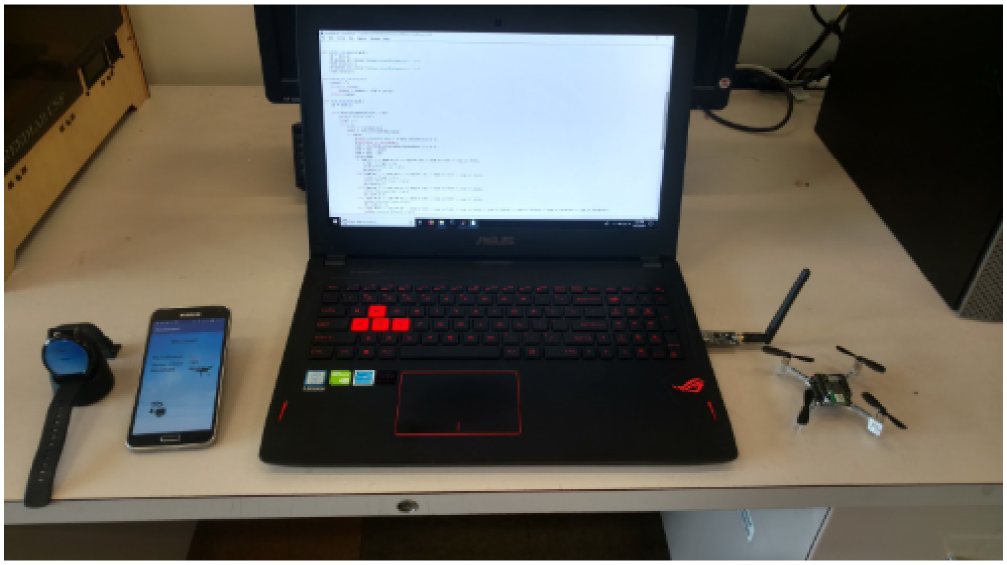

Figure 10. Communication equipment setup.

control (i.e., altering the reference scaling and formation orientation) were tested to create a V-formation and travel through a narrow passage, without global information exchange. This was accomplished when the user sent inputs for command tracking, scaling, and orientation, via a workstation, to the leader. The proposed architecture tested allowed for information exchange between agents. In addition, an application involving sending scaling factors, via arm motion with a smartwatch, to the leader was tested and verified. Furthermore, a small time delay was seen when performing the experiments in regards to the position of each Crazyflie. Future work will include adding an additional layer of consensus for error to correct for the time delay and extending the practical application to allow for increased human interaction when giving commands to the formation.

References

[1]D. Tran and T. Yucelen, "Control of multiagent formations: A multiplex information networks-based approach," in *ASME Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2015, pp. V003T37A002–V003T37A002.

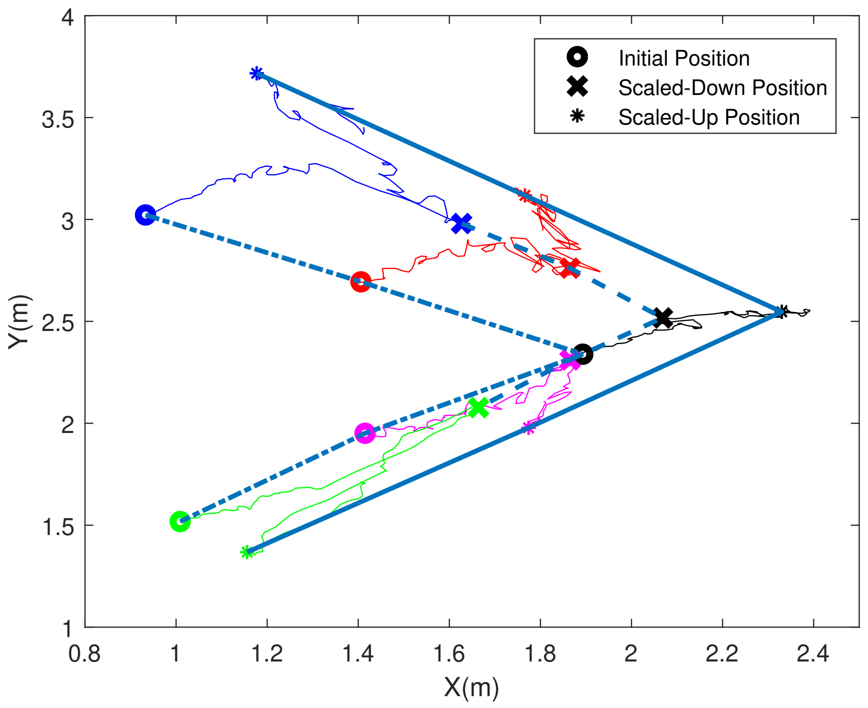

Figure 11. The V-formation shown varying in scale.

[2] D. Tran, T. Yucelen, and E. Pasiliao, "Multiplex information networks for spatially evolving multiagent formations," in *American Control Conference*. IEEE, 2016, pp. 1912–1917.

[3] ——, "Formation control with multiplex information networks," *IEEE transactions on control systems technology*, 2018 (under review).

[4] "Crazyflie 2.0," https://wiki.bitcraze.io/projects:crazyflie2:index, [Accessed 07-June-2018].

[5] "Loco positioning deck," https://wiki.bitcraze.io/projects:lps:deck, [Accessed 07-June-2018].

[6] "Flow expansion deck," https://wiki.bitcraze.io/projects:crazyflie2:expansionboards:flow, [Accessed 07-June-2018].

[7] "Moto 360 1st gen," https://support.motorola.com/ca/en/products/wearables/moto-360, [Accessed 13-November-2018].