ASEAN Engineering Journal

Full Paper

SWARM UAV IMPLEMENTATION USING RADIO LOCALIZATION ON GPS DENIED AREAS

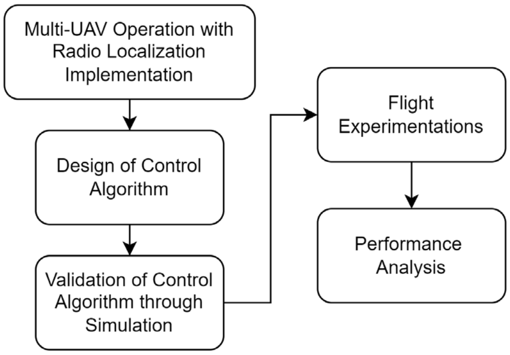
Timothy Scott Chu[a]*, Alvin Chua[a], Marc Say[b], Edwin Sybingco[b], Maria Antonette Roque[b]

[a]Department of Mechanical Engineering, De La Salle University, 2401 Taft Ave, Malate, Manila, 1004 Philippines
[b]Department of Electronics and Computer Engineering, De La Salle University, 2401 Taft Ave, Malate, Manila, 1004 Philippines

Article history
Received
21 October 2021
Received in revised form
16 April 2022
Accepted
29 April 2022
Published online
31 August 2022

*Corresponding author
timothy.chu@dlsu.edu.ph

Graphical abstract


Multi-UAV Operation with Radio Localization Implementation
Design of Control Algorithm
Validation of Control Algorithm through Simulation
Flight Experimentations
Performance Analysis

Abstract

Swarming is a rapidly growing idea that is being implemented in UAV applications. Its effectiveness and efficiency in finding solutions or executing the desired task served as the main motivation for this study. Localization techniques are vital for swarm implementation and deployment since it one of the main determining factors in its performance. Vision systems have been widely used for localization; however, it may be costly as it requires multiple appropriate cameras. Another localization technique, which is explored in this research, is radio localization. This localization employs Ultrawide-Band radios to communicate with each other to return a target's position with respect to several reference points. The study presents a new collaborative UAV implementation deployed using radio localized systems for harsh or unknown environments. The study used the Loco Positioning System operating on the Time Difference of Arrival protocol to maneuver two UAVs in a workspace. The study determined how well the system can execute the desired flight path and the performance of the system in keeping the set distance between UAVs to avoid possible collisions. Results of the study showed that the proposed implementation was successful in maneuvering the UAVs flying 0.3 m apart.

Keywords: Crazyflie, Loco Positioning System, Radio Localization, Swarm Drone, Unmanned Aerial Vehicles

1.0 INTRODUCTION

Unmanned Aerial Vehicle (UAV) Swarming, the utilization of two or more UAV units in a control system to fulfill the desired function, is drawing the attention of various researchers due to its known capacity to obtain data as efficiently as possible in a given implementation. The research [1] presents a use case where a swarm drone implementation was used for geomatic data collection. The main objective of their study was to implement a swarm control system that can adapt to the dynamically changing resolution of a drone's camera for surveying. This allowed the implementation of a swarm that was comprised of drones with different camera quality, to effectively collect geomatic data. The research works [2-4] highlighted the effectiveness of UAV drones in rescue missions and in providing services on dangerous terrains. The integration of swarming was known to benefit professionals in executing their responsibilities in the mentioned fields. On top of these applications, the study [5] highlighted that UAV swam are can also be applied in areas of security and surveillance, disaster management, and even leisure. A common leisure application for UAV swarms is the frequently witnessed light shows that utilize drones as a green alternative to fireworks. As the number of potential applications of Swarm UAV increases, a significant effort in developing efficient control systems to aid in maneuvering multiple UAVs, while requiring the user to remotely control at most a single drone becomes more and more necessary [6-10]. A critical factor in the control system for swarm implementation is its localization system. Localization is the technique where external sensors, such as satellites, vision cameras, or radio components, are integrated into the system to detect and locate the aerial drone and represent its location in XYZ coordinates of a defined workspace, or longitudes and latitudes. *Global Positioning System (GPS)* employs satellites that detect a tagged UAV and reflects its position in the context of longitudes, latitudes; however, this system poses limitations as it becomes inaccurate when the aerial drone operation is conducted in areas where the direct line of sight between the drone and satellite is obstructed, this is also known as GPS denied environments. To address this limitation, indoor localization techniques utilizing various vision cameras were developed [11]. *Vision Localization* often makes use of several cameras strategically positioned around a workspace; and reflective markers were placed on the objects of interest, in this context, UAVs. This system aims to capture different angles of the aerial drone during operation and collate the data to accurately return the position data of each drone. This would allow UAVs to determine their position in a workspace and coordinate with other UAVs to adapt accordingly and avoid collision [12].

J. Priess et. al. in [13] explored a swarm implementation called the Crazyswarm. The study made use of up to 49 Crazyflie drones, all equipped with reflective markers. Using a Vicon Tracker, the study was able to maneuver all 49 drones in hovering at varying heights and in performing a different range of motions. The proponents utilized vision localization in their control system, allowing the system to generate and compare each subsequent image frame to track the movement of every Crazyflie. This also allowed each UAV unit to autonomously adjust itself to avoid a potential collision or when the system detects that the drone was not where it was supposed to be.

Buffi et. al. [14] explored an RFID approach for UAV localization. The study focuses on the SARFID localization technique, which tracks the movement of an antenna, attached to the object-of-interest, with respect to several static tags. The experimental setup was designed to be conducted in an outdoor scenario with about 25 tags spaced 2 meters apart from each other in a grid formation. The UAV was operated remotely and was tasked to fly over the static tags; and, the flight part obtained from the SARFID localization is compared to the flightpath obtained from GPS. The study obtained satisfactory results and noted that the system can also be implemented for Unmanned Ground Vehicles (UGV).

Another localization technique is called *Radio Localization* wherein the technique employs radio signals and the time-of-flight concept. Chu et. al conducted a study [15] to analyze the performance of the Loco Positioning System, a radio localization technique. A typical setup of radio localized implementations would require several anchors and tags. Anchors are used in setting a workspace, consequently serving as the boundary as well. A tag served as the object of interest and was mounted on the UAV. Both the anchors and deck communicate with each other by sending data packets through radio signals, allowing the UAV drone to calculate its position with respect to the anchor setting and placement. The mentioned study offered multiple setup cases and their effectivity which aided this research in its experimental setup.

This study is motivated and aimed to explore the potential of radio localization for swarm drone implementation, as it is an affordable option compared to a vision system. In addition, radio localization that utilizes Ultra-wideband (UWB) radios are substantially efficient in GPS denied environments for short and immediate-range localization [16-17]. This paper presents an experimental setup for swarming in GPS denied environments as well as obtaining a performance analysis on the system implementation utilizing radio localization, specifically the Loco Positioning System. This experimental setup can serve as a basis for similar control systems on multiple air or land drone implementation that utilize radio localization.

2.0 THEORETICAL CONSIDERATIONS

Loco Positioning System

The Loco Positioning System is a radio localization implementation developed by the proponents at Bitcraze for the Crazyflie drone. It is a modular expansion to the Crazyflie drone that consists of Loco Positioning Nodes and Loco Positioning Decks. The Loco Positioning Nodes primarily serve as the anchors for boundary setting, but they can also be configured to act as a Loco Positioning Deck. The Loco Positioning Deck serves as the tag of the system and is attached to the Crazyflie drone to mark it as the object of interest. The two components are based on the Decawave DWM1000 chip which is an Ultra-wideband (UWB) radio. UWB radios are capable of transmitting data through high bandwidth, around more than 500 MHz, while requiring low energy and are usually suitable for relatively short distances. A sufficient LPS implementation is usually comprised of six nodes, to form a cubic boundary, and a single deck mounted on the UAV as shown in Figure. 1. The system employs estimators such as the Kalman Estimator to process the obtained information, in the form of communicated data packets between the node and the deck; and, returns an estimated position of the LPS Deck mounted Crazyflie within the design boundary. The system operates mainly on two protocols, the Two-Way Ranging (TWR) protocol and the Time Difference of Arrival (TDoA) protocol; and, both protocols are supplemented with the Time-of-Flight concept to calculate the distance of the tag with respect to an anchor [18]. The research focuses on utilizing the latter protocol as it is less computationally intensive and is more suited for swarm applications [19].

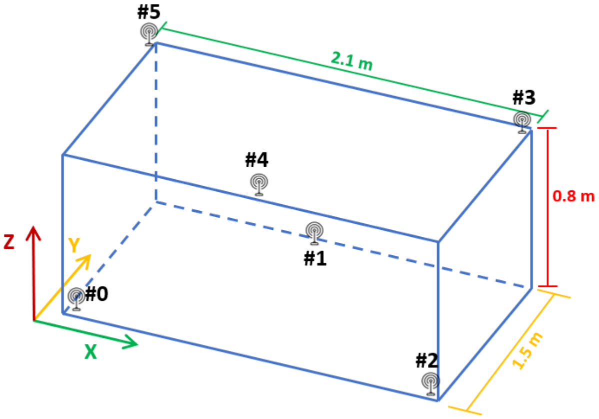


Figure 1 Loco Positioning System with 6 Anchors

Time Difference of Arrival (TDoA) Protocol

The Time Difference of Arrival protocol of the LPS can be divided into two categories Time Difference of Arrival 2 (TDoA2) and Time Difference of Arrival 3 (TDoA3) [19]. The general configuration for this protocol is that the LPS Nodes continually transmit data packet signals while the LPS Decks are solely tasked to receive and process the incoming data. The whole process is less computationally intensive and is faster than the traditional process which requires an exchange of messages between components. Computationally, the TDoA protocol can be generally defined as the difference between the time data is transmitted (T_x) and the time data is received (R_x). However, in a real system composed of multiple anchors, the LPS Deck also considers data of other LPS Nodes into the computation which makes the equation a bit more complex. Figure. 2 represents a simple but complete system implementation with two LPS Nodes marked as Anchor #0 and Anchor #1. ANCHORS 0 and 1 both transmit data packets, represented by T_x0 and T_x1 respectively, to be received by the tag, represented as R_x0 and R_x1 respectively. This is observed in E2 and E3 of Figure 2. The TDoA of this system is computed by getting the overall difference between the difference in received data packets (ΔR_x) and the difference in transmitted data packets (ΔT_x), shown in equation 1

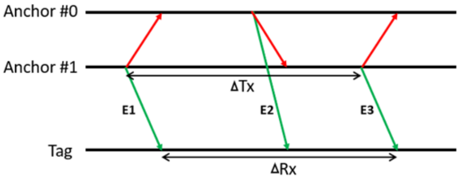


Figure 2 Time Difference of Arrival Protocol

$$TDoA = \Delta R_x - \Delta T_x \qquad (1)$$

To address the potential clock drift error in the system, a Corrective Factor (C.F.) is introduced. The C.F. is obtained by observing if the interval of the transmission and reception of data packets from a single ANCHOR is consistent, hence is expressed in equation 2 and visually represented by E1 and E3 in Figure 3. Considering the C.F. into the calculation of TDoA is represented in equation 3 [21].

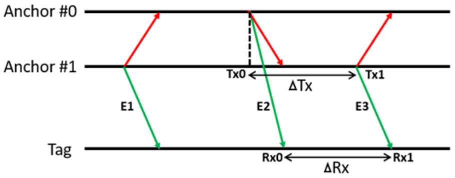


Figure 3 Time Difference of Arrival with Corrective Factor

$$C.F. = \Delta R_x 1 / \Delta T_x 1 \qquad (2)$$

$$TDoA = \Delta R_x - (C.F.* \Delta T_x) \qquad (3)$$

where:

T_x – Time of transmission of data packet

R_x – Time of receipt of data packet

ΔR_x – Time difference between exchanges for received data packets

ΔT_x – Time difference between exchanges for transmitted data packets

E – Exchange of data packet from anchor to tag

$TDoA$ – Time of complete exchange of data packets

As previously mentioned, the LPS TDoA protocol has two subcategories, TDoA2, and TDoA3. The TDoA2 is a stable implementation provided that the user strictly follows the configuration set by Bitcraze, utilizing a maximum of eight ANCHORS into the system. This protocol offers results that are comparable to the TWR protocol of the system. TDoA3 operates similarly to TDoA2, the main difference between the two is the time for transmitting data packets. While TDoA2 follows a consistent interval for data transmission, TDoA3 randomizes the interval in data transmission as an attempt to further reduce the computational load of the system. This allows the TDoA3 protocol to accompany more than eight anchors, consequently allowing the possibility of up-scaling of the flight space of the UAV drone.

Control Algorithm

The logic of the program of the swarm system is represented in Figure 4. The process begins with setting the initial parameters of the system. The parameters are the initial UAV positions, specifying the number of UAVs in the system, maintaining distances, specifying the reference UAV, and trajectory of the reference UAV. Flight operations can be conducted once the initial parameters are set. Somewhere along the flight trajectory, the follower UAVs may be positioned on the relative left or relative right of the reference UAV. During operation, the follower UAV would consistently check its position with respect to the reference UAV and adjust accordingly to maintain the indicated distance to avoid collision with other UAV units. The simple program implementation is robust and could be implemented with at least two UAV units and can expand to more units. The program may also be applied to Unmanned Ground Vehicles (UGV) applications provided that it utilizes radio localization.

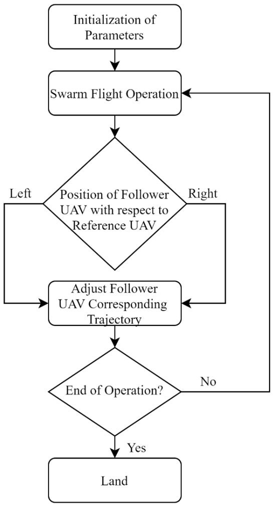

Initialization of Parameters
Swarm Flight Operation
Left
Position of Follower UAV with respect to Reference UAV
Right
Adjust Follower UAV Corresponding Trajectory
End of Operation?
No
Yes
Land

Figure 4 Control Algorithm Flowchart

3.0 METHODOLOGY

Materials

The Crazyflie, shown in Figure. 5a is a small, light, but robust UAV developed by Bitcraze. With its size and weight, the UAV drone possesses a small momentum force making it relatively harmless upon collision. The Crazyflie is equipped with the minimum requirements for flight; however, it is also designed to be modular where expansion modules can be easily installed to the

Crazyflie. The Crazyflie is an open-source drone, making it a popular choice for researchers to serve as a prototype in different applications [22]. This study employs two units of Crazyflie drones in the system. Additionally, Loco Positioning Node and Deck, shown in Figure. 5b and c respectively, are used in this research.

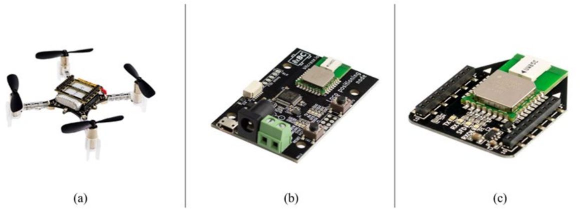


Figure 5 (a) Crazyflie Unit, adapted from [22], (b) Loco Positioning Node, adapted from [23], (c) Loco Positioning Deck, adapted from [24]

Research Methodology

Figure. 6 presents the three stages of the research methodology of this study. To start, researchers designed a suitable workspace for test flight based on the recommendations in [15]. The next step was to execute a simulation test to ensure that the developed program could provide the desired outcome. Finally, actual flight tests were conducted while gathering the positional data to determine if the actual flight path resembles the simulated flight path, and how well the distance between two UAVs was maintained.

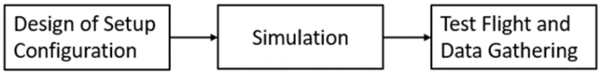


Figure 6 Research Methodology

Design of Setup and Configuration

Figure. 7 shows the setup configuration for the LPS. A total of eight anchors were used positioned strategically to form a rectangular box. The study [15] highlighted that more anchors and large displacements between anchors offer better results. Thus, the distance between anchors along the X–axis is 2.1 m and 1.5 m along the Y–axis. The distance along the Z–axis is set to be 0.8 m. This setup creates an enclosed experimental volume where the Crazyflie units, denoted as CF#1 and CF#2, may fly around, and its trajectory is recorded through radio localization which will be compared to a theoretical setup to obtain the system's performance. Currently the experimental setup makes use of two Crazyflie units, however, additional units can be added into the system by simply specifying the initial position of the additional units.

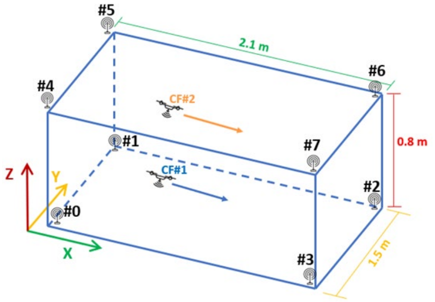


Figure 7 Experimental Setup

Simulation

Python programming was the main programming language of choice together with Robot Operating System (ROS). The logic of the program enabled the second UAV, from now on referred to as the follower UAV, to generate its own flight path based on the flight path of the first UAV, the reference UAV. Consequently, the follower UAV should fly and create a path similar to the reference UAV, albeit larger or smaller depending on its initial position. The goal of the follower UAV was to maintain a specified gap from the reference UAV while following it. Mainly two flight paths were used in this research, a linear flight path, and a box flight path. On the linear flight path, the goal was to determine how well the follower UAV maintains a 0.5 m distance from the reference moving along the X–axis starting from 0.3 m then to 0.7 m and finally 1.3 m before landing. This is shown in Figure. 8 where the follower UAV was positioned on (0.3, 0.95) and the reference drone positioned on (0.3, 0.45) along the X and Y axes. On the box path, the reference UAV was positioned along (0.3, 0.3) on the X and Y axes while the follower UAV was positioned on (0.3, 0.6). The objective of the follower UAV was to maintain a 0.3 m gap from the reference UAV. The follower UAV also should mimic the same flight path; however, it would need to adjust itself to lag from the reference UAV to avoid potential collision or crossing of flight paths. Figure. 9 shows the expected flight path for both UAVs forming a box path. This also shows that the follower UAV has made a smaller box path compared to the reference UAV. For simulation purposes only, given that the reference UAV's movement along the Y-axis is opposite from the initial setup. The follower UAV would consequently go beyond the reference UAV forming a larger box. This simulation example is shown in Figure. 10.

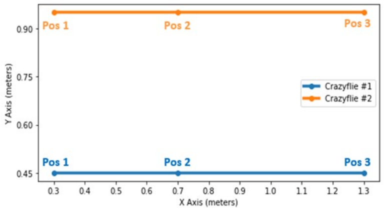


Figure 8 Linear Flight Path Simulation

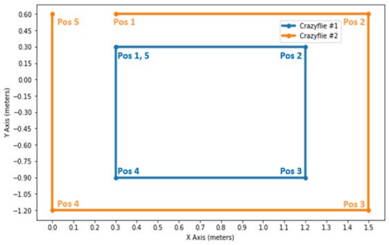


Figure 9 Box Path Simulation [Case 1]

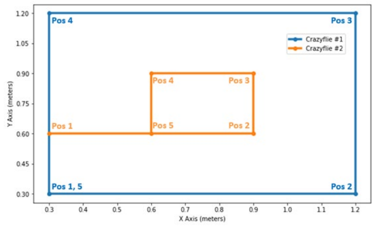


Figure 10 Box Path Simulation [Case 2]

Actual Flight Tests

Figure. 11 shows the experimental setup used in the test flight. Both UAVs were tasked to execute the mentioned scenarios except for the Box Path – Case 2. Positional data of the Crazyflie units in the flight space were obtained every 0.1 seconds and presented in X, Y, Z meter values. Three trials are done for each scenario and the average data is obtained and plotted. This provided a visual of the UAV's actual performance in executing the desired flight path as well as validating the viability of the algorithm in swarm implementation. Additionally, position values in each interval for both UAVs were compared to see how well the drone maintained the desired distance between the two Crazyflie units.

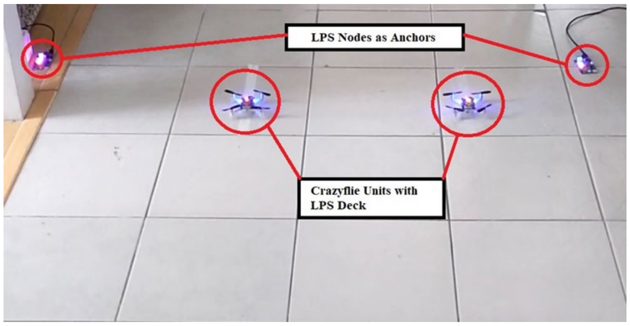


Figure 11 Actual Experimental Setup

4.0 RESULTS AND DISCUSSION

Figures 12 and 13, show the actual flight path executed by the UAVs for linear flight path and box flight path respectively. The blue diamond symbols refer to the positional data of Crazyflie #1, reference UAV across time, while orange boxes refer to Crazyflie #2, follower UAV's positional data. Figure. 11 represents scenario #1 Linear Flight Path, and it was observed that both UAVs were able to execute the desired movement of flying from 0.3 m to 1.3 m along the X-axis. The caveat, however, was that the numerical data showed that the UAVs have overshot the target by a maximum of 0.24 m towards the end of the experiment. Theoretically, the distance to maintain by Crazyflie #2 is 0.5 m along the Y-axis. Taking the difference of the Y coordinates between both UAVs and averaging of the result, it was found that the drone was able to maintain an average distance of 0.42 m, which is 0.08 m short of the theoretical distance, as shown in Table 1. In scenario #2 shown in Figure. 12, the plotted actual positional data formed a pattern resembling that of the simulation scenario in Figure. 10. Both Crazyflie units started at the same starting point but Crazyflie #2 formed a smaller box pattern compared to the box formed by Crazyflie #1. Referring to the same figure, it is clear that the actual flight path has overshot the 1.2 m mark along the X and Y axes. Numerical data confirms that actual test flights overshot the X-axis target by a maximum of 0.21 m and a maximum of 0.13 m for the Y-axis. Generally, the follower UAV must maintain a 0.3 m distance on both X and Y axes with respect to the reference UAV. Taking the average of the differences of each point for both UAVs, it was found that Crazyflie #2 maintained an average distance of 0.27 m. In both scenarios, the program was able to maneuver the drone in conducting the desired flight path pattern. Scenario #1 tasked the UAVs to move from 0.3 m to 1.3 m along the X-axis whilst maintaining a 0.5 m gap between UAVs, and Scenario #2 performing a box path flight patterns with the same set of coordinates mentioned in Case 1 under Simulations section. Though comparing the theoretical flight path setup against the actual values, it was observed that the actual flight overshot the target by a maximum of 0.24 m for the first scenario, and a maximum of 0.21 m along the X-axis and 0.13 m along the Y-axis on the second scenario. Evaluating the acceptability of these overshoots mainly depends on the context of the application. For implementations with large areas, spanning for 5 meters or more, this should be an acceptable margin. It should be noted that the experimental setup was able to maintain a good distance between the UAV drones, possessing variances of 0.08 m for the first scenario and 0.03 m for the second scenario. It is also worth noting that the LPS implementation was to manage the Crazyflies well given that both units needed to maintain a 0.3 m gap.

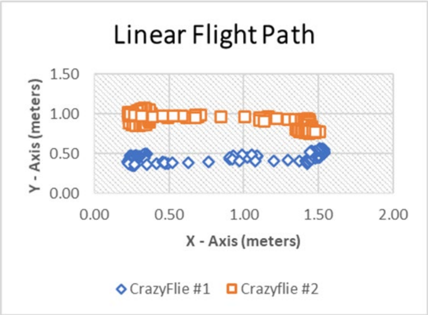


Figure 12 Actual Linear Flight Path Plot

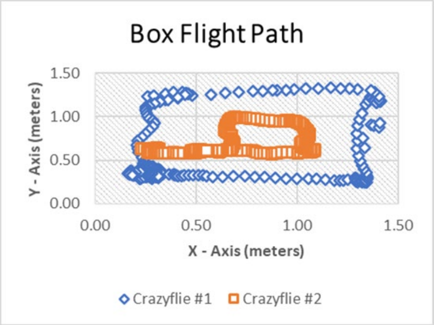


Figure 13 Actual Box Flight Path Plot

Table 1 Summary of Variances per Case

Variable	Linear Flight Path	Box Flight Path
X Theoretical	-	0.50 m
Average X Actual	-	0.42 m
Variance	-	0.08 m
Error (%)	-	16%
Y Theoretical	0.30 m	0.30 m
Average Y Actual	0.27 m	0.27 m
Variance	0.03 m	0.03 m
Error (%)	10%	10%

5.0 CONCLUSION

This research focused on swarm drone implementation where a follower UAV calculated its flight path with respect to the flight path of the reference UAV. The follower UAV mimics the flight path of the reference UAV; however, it creates a scaled version of the flight path depending on its initial position in the system. The research focused also on implementing the system with radio localization, specifically using LPS developed by Bitcraze. The LPS uses Ultrawide-band radios with accuracy in the range of 0.1 m. Simulations offered 3 scenarios of different flight paths. This was done using python programming and ROS to initially check whether there were potential faults in the implementation, before the actual flight tests. An experimental setup was presented in this study for verification. This involved conducting simulation tests and actual flight tests, where the latter possessed 2 flight path scenarios, linear and box flight paths. Positional data of the flight tests were obtained and translated in terms of X, Y, and Z measures in meters. Obtained data showed that the UAVs were successful in moving along the desired flight path while maintaining a gap between UAVs. Upon comparison between the theoretical and actual flight path, a degree of similarity was observed, confirming the viability of the system. The results of the experiment showed that the UAVs kept a sufficient distance apart, deviating about 0.03 m to 0.08 m from the design parameter, consequently validating the 0.1 m accuracy range of the LPS system. The presented experimental setup could be used as a reference for future swarming research, prototyping, or implementation using both the Crazyflie and the LPS system since the setup also presented a performance analysis of the system in actual test flights.

In the future, the researchers plan to expand this study by implementing more UAV units into the system and performing different flight patterns to further evaluate the performance of radio localization. This also aims to validate the expandability and the viability of the developed control system for GPS denied areas. Further testing can be done to determine the versatility of this control system by implementing it to UGV swarm implementations.

Acknowledgement

The proponents of this research would like to extend its gratitude to University Research and Coordination, De La Salle University – Manila and DOST-PCIEERD for providing resources which allowed the proponents to conduct the study.

References

[1] Almeida, M., Hildmann, H., & Solmaz, G. 2017. Distributed UAV-swarm-based real-time geomatic data collection under dynamically changing resolution requirements. The International Archives of Photogrammetry, *Remote Sensing and Spatial Information Sciences,* 42: 5. DOI: 10.5194/isprs-archives-xlii-2-w6-5-2017

[2] Apvrille, L., Tanzi, T., & Dugelay, J. L. 2014. Autonomous drones for assisting rescue services within the context of natural disasters. In 2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS) 1-4. IEEE. DOI: 10.1109/URSIGASS.2014.6929384.

[3] Montufar, D. I., Munoz, F., Espinoza, E. S., Garcia, O., & Salazar, S. 2014. Multi-UAV testbed for aerial manipulation applications. In 2014 I*nternational Conference on Unmanned Aircraft Systems* (ICUAS) 830-835. IEEE. DOI: 10.1109/ICUAS.2014.6842329.

[4] Karaca, Y., Cicek, M., Tatli, O., Sahin, A., Pasli, S., Beser, M. F., & Turedi, S. 2018. The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations. *The American Journal Of Emergency Medicine*, 36(4): 583-588.DOI: 10.1016/j.ajem.2017.09.025.

[5] Tahir, A., Böling, J., Haghbayan, M. H., Toivonen, H. T., & Plosila, J. 2019. Swarms of unmanned aerial vehicles—a survey. *Journal of Industrial Information Integration,* 16, 100106. DOI: 10.1016/j.jii.2019.100106.

[6] Cummings, M. L., Bertucelli, L. F., Macbeth, J., & Surana, A. 2014. Task versus vehicle-based control paradigms in multiple unmanned vehicle supervision by a single operator. *IEEE Transactions on Human-Machine Systems*, 44(3), 353-361. DOI: 10.1109/THMS.2014.2304962.

[7] Dim, C., Nabor, F., Santos, G., Schoeler, M., & Chua, A. 2019. Novel experiment design for unmanned aerial vehicle controller performance testing. In IOP Conference Series: Materials Science and Engineering 533(1): 012026. *IOP Publishing*. DOI: 10.1088/1757-899x/533/1/012026.

[8] Piquero, J. L. P., Delica, V. K., Orquia, A. L., Reynaldo, E. M., Ilao, J., Roque, M. A., ... & Jayakody, H. 2019. A new sliding mode controller implementation on an autonomous quadcopter system. *International Journal of Automation and Smart Technology*, 9(2), 53-63.

[9] Mueller, M. W. 2018. A dynamics-agnostic state estimator for unmanned aerial vehicles using ultra-wideband radios. In Dynamic Systems and Control Conference 51913: V003T36A002. American Society of Mechanical Engineers.

[10] Fahandezh-Saadi, S., & Mueller, M. W. 2018. An algorithm for real-time restructuring of a ranging-based localization network. In 2018 International Conference on Unmanned Aircraft Systems (ICUAS) 236-242. IEEE. DOI: 10.1109/ICUAS.2018.8453294.

[11] Badshah, A., Islam, N., Shahzad, D., Jan, B., Farman, H., Khan, M., ... & Ahmad, A. 2019. Vehicle navigation in GPS denied environment for smart cities using vision sensors. C*omputers, Environment and Urban Systems*, 77: 101281.

[12] Ahmed, H., & Glasgow, J. 2012. Swarm intelligence: concepts, models and applications. School Of Computing, Queens University Technical Report 2012-585.

[13] Preiss, J. A., Honig, W., Sukhatme, G. S., & Ayanian, N. 2017. Crazyswarm: A large nano-quadcopter swarm. In 2017 IEEE International Conference on Robotics and Automation (ICRA). 3299-3304. IEEE.

[14] Buffi, A., Nepa, P., & Cioni, R. 2017. SARFID on drone: Drone-based UHF-RFID tag localization. In 2017 IEEE International Conference on RFID Technology & Application (RFID-TA) 40-44. IEEE. DOI: 10.1109/RFID-TA.2017.8098872.

[15] Chu, T., Chua, A., Sybingco, E., & Roque, M. 2019. A Performance Analysis on Swarm Drone Loco Positioning System for Time Difference of Arrival Protocol. International Journal of Engineering and Advanced Technology, 9(2): 1475-1485. DOI: 10.35940/ijeat.b3480.129219

[16] Chen, H., Xian-Bo, W., Liu, J., Wang, J., & Ye, W. 2020. Collaborative multiple UAVs navigation with GPS/INS/UWB jammers using sigma point belief propagation. *IEEE Access,* 8: 193695-193707. DOI: 10.1109/ACCESS.2020.3031605.

[17] Sivaneri, V. O., & Gross, J. N. (2017). UGV-to-UAV cooperative ranging for robust navigation in GNSS-challenged environments. *Aerospace science and technology*, 71: 245-255. DOI: 10.1016/j.ast.2017.09.024.

[18] Dädeby, S., & Hesselgren, J. 2017. A system for indoor positioning using ultra-wideband technology (Master's thesis). Dept. of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden.

[19] Loco Positioning system. Bitcraze. 2020. Retrieved June 27, 2020, from https://www.bitcraze.io/documentation/system/positioning/loco-positioning-system/.

[20] Loco Positioning TDoA Principles. Bitcraze. 2020. Retrieved September 17, 2020, from https://www.bitcraze.io/documentation/repository/lps-node-firmware/2020.09/functional-areas/tdoa_principles/.

[21] TDoA2 vs TDoA3. Bitcraze. 2020. Retrieved September 17, 2020, from https://www.bitcraze.io/documentation/repository/lps-node-firmware/master/functional-areas/tdoa2-vs-tdoa3/.

[22] Crazyflie 2.1. Bitcraze. 2019. Retrieved September 10, 2019, from https://www.bitcraze.io/crazyflie-2-1/.

[23] Loco Positioning Node. Bitcraze. 2019. Retrieved September 10, 2019, from https://www.bitcraze.io/loco-pos-node/.

[24] Loco Positioning Deck. Bitcraze. 2019. Retrieved September 10, 2020, from https://www.bitcraze.io/loco-pos-deck/.