Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi
Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering

Basılı/Printed ISSN: 1302-9304. Elektronik/Online ISSN: 2547-958X

Web Based Control of Multiple UAVs Using Self-Healing Network Architecture in GPS denied Environment

GPS'siz Ortamlarda Kendini Onaran Ağ Mimarisi Kullanan Sürü Uçuşunun Web Tabanlı Kontrolü

Onur Keskin [1]*, Zeki Kıral [2]

[1] Dokuz Eylül University, The Graduate School of Natural and Applied Sciences, Department of Mechatronics Engineering, 35160 Tınaztepe, Buca-İzmir, TÜRKİYE
[2] Dokuz Eylül University, Faculty of Engineering, Department of Mechanical Engineering, 35390, Tınaztepe, Buca-İzmir, TÜRKİYE

Sorumlu Yazar / Corresponding Author *: onur.keskin@deu.edu.tr

Geliş Tarihi / Received: 16.11.2019 *Araştırma Makalesi/Research Article*
Kabul Tarihi / Accepted: 27.12.2019 *DOI:10.21205/ deufmd.2020226528*
Atıf şekli/ How to cite: KESKİN, O., KIRAL, Z. (2020). Web Based Control of Multiple UAVs Using Self-Healing Network Architecture in GPS Denied Environment. DEUFMD 22(65), 625-637.

Abstract

Unmanned aerial vehicles (UAVs) become very popular in the last years with the help of increasing computing power per area and per cost. While UAVs with a global positioning system (GPS) can easily operate to fly autonomously, this and such sensors' data cannot always be trusted. And most of the cases for small scale UAVs, we cannot use these kinds of sensors because of cost, complexity, and weight. Safely and reliably operating close to unknown indoor or GPS-denied environments requires improving UAVs' sensing, localization, and control algorithms. To solve and to improve for a UAV is one problem; extending it to multiple UAVs is another problem. We study and develop a framework for multiple UAVs to command and control from web-based front-end. In our experiments, a quadcopter platform called AR Drone from Parrot Inc. used because of its open-source API and kernel-level arrangements. Test flights inside a warehouse validate the framework is capable of control one or more quadcopters in a given lattice-based path from a web browser. UAVs are capable of localizing its position by using IMU, front, and bottom cameras. All UAVs interconnected to each other and controller computer through a self-healing network, so if one or more quadcopter fails because of lack of battery or any other circumstances, the rest of the group continues its mission. Experiments are also expanded to outdoor to demonstrate rooftop trips for vehicles in convoy and also reconnaissance missions, especially for Mine-Resistant Ambush Protected (MRAP) vehicles.

Keywords: *Unmanned Aerial Vehicle, Quadrotor, Multi-Agent Systems, Self-Healing Network, Swarm Flight*

Öz

İnsansız hava araçları (İHA) gelişen işlem gücünün hem alan hem de maliyet bazında artış göstermesi ile olukça yayınlaşmıştır. Küresel konumlandırma sistemi (GPS) kullanan İHA'ların otonom uçuşları, kullanmayanlara göre çok daha uygulanabilirdir. Ancak GPS ve benzeri algılayıcılardan alınan veriler her zaman güvenilir değildir. Ayrıca küçük ölçekli İHA'larda bu tarz algılayıcılar artan maliyet, karmaşıklık ve ağırlık seviyelerinden dolayı kullanılması uygun olmamaktadır. Özellikle daha önceden bilgi edinilmemiş iç mekanlarda veya GPS ve benzeri algılayıcıların çevre koşullarından dolayı kullanılamadığı alanlarda güvenli ve güvenilir bir şekilde çalışabilmek için İHA'ların algılama, konumlandırma ve kontrol algoritmalarını geliştirmek gerekmektedir. Tek bir İHA için adreslenmiş bu sorunları çözmek bir problem iken, bunu birden çok İHA ile birlikte yapabilmek ayrı başlıca bir problemdir. Bu çalışmada, web tabanlı bir kullanıcı arayüzü üzerinden birden çok İHA'yı kumanda ve kontrol edebilme yeteneğine sahip yazılım çerçevesi geliştirilmiştir. Test ve deneylerimiz esnasında, açık kaynaklı uygulama programlama arayüzü (API) ve gömülü yazılımına çekirdek seviyesinde müdahale imkanı olan Parrot firmasına ait ARDrone isimli dört rotorlu döner kanatlı uçan platformlar kullanılmıştır. Fabrika depo bölgesinde, web tarayıcısı üzerinden verilen ızgara (lattice, grid) tabanlı yol planı ile icra edilen test uçuşları; geliştirilen yazılım çerçevesinin bir ve daha fazla dört rotorlu hava araçlarına başarılı görevler yaptırdığı da doğrulamıştır. Testlerde kullanılan İHA'ların kendi konumlarını çalışma bölgelerinde bulabilmesi için üzerlerinde bütünleşik olarak yer alan ataletsel ölçüm birimi (IMU), ön ve alt kamera kullanılmıştır. Uçuş görevindeki tüm İHA'lar birbirlerine, ana ağ geçidi olan İHA batarya veya farklı bir sorun ile ulaşılamaz olması durumunda bile iletişimlerine ve görevlerine devam edebilecekleri, kendini onaran ağ mimarisi ile bağlıdırlar. Konvoydaki araçlar ve özellikle de mayına dayanıklı pusu korumalı (MRAP) araç üzerlerinden gezi ve keşif amaçlı görevleri göstermek için deneyler dış mekanlara da genişletilmiştir.

Anahtar Kelimeler: *İnsansız Hava Aracı, Döner Kanatlı, Çok Ajanlı Sistemler, Kendini Onaran Ağ, Sürü Uçuşu*

1. Introduction

Unmanned aerial vehicles (UAVs) encountered an increasing number of applications, mostly military and public [1, 2]. They offer major advantages when used for aerial information, surveillance, target acquisition, and reconnaissance (ISTAR) in complex and dangerous environments. Lower downside risk and higher confidence in mission success are two strong motivators for continued expansion of UAVs. Many other technological, economic, and political factors have also contributed to the development and deployment of UAVs. The main UAV applications are defense-related, and future military scenarios drive the main investments. However, a civil UAV market predicted to emerge over the next decade, starting with government organizations requiring surveillance and inspection systems, such as coast guards, border patrol organizations, emergency services, and police.

In all these situations, the global positioning system (GPS) acts as a main sensory input. While GPS proves itself in different environments and conditions, its data cannot always be got or trustworthy [3]. So, using UAVs at indoor or any GPS-denied environments gain a significant amount of research and development [4]. And also, sometimes, redundancy becomes a vital feature to counteract misleading information [5]. To overcome and to replace this high-quality positioning output, first substitute sensor is a camera and various kinds of computer vision algorithms [6]. Camera, the sensor itself, maybe low-cost hardware, but its computational cost must be considered. It consumes much more energy and has higher computational complexity than GPS. Second alternative is signal triangulation techniques, which is the same approach with GPS satellites [7]. Third technique based on an internal measurement unit and time of flight (ToF) sensors such as sonar, radar, or LIDAR. With the help of sensor fusion, simultaneous localization and mapping (SLAM) algorithms can calculate, construct, and update unknown environments. Although all these alternatives are lack of absolute positioning, the challenge is to develop reliable position measurement with inspiring from nature.

"Divide and conquer" approach can also be applied for guiding unmanned vehicle which uses computer vision algorithms. The smallest meaningful piece of the vision system is a pixel. When we project this unit on a vehicle scale, it defines the grid. Especially in aerial vehicles, using a perfect square as a measurement unit and plan all the missions for it, may mislead your UAV [8]. So that instead of grids, lattices can also be used with various kinds of convex geometries to perform path planning and path following [9].

Communication is a critical feature in many systems. And it is much more critical when you deal with a flight vehicle. Although UAV flight by itself, it may stick somewhere when communication is lost even in one millisecond or less. To maintain a strong link between controller and UAV, redundancy techniques, high transmission power antenna, cellular, or satellite communications may be used [10]. In recent years, artificial intelligence techniques also studied to build self-determined mission planning algorithms. But in the end, in many real-life scenarios, you have to maintain a connection to start a mission, track mission steps, or gather results of the task. Because of the current battery technologies, UAV that uses electric transmission has under an hour flight time with a single power source. In realistic missions, flight time must be higher, and to solve this bottleneck, on-air charging or battery swapping is also considered. And for an alternative approach, tethered UAVs are also both in the consumer and military market for "unlimited" flight time. We can consider communication with two nodes. But in a multiple robot scenario, it became more critical and complicated. Using a router infrastructure locks your moving vehicles to a specific area, and you do not have any redundancy. Also, you have a third device that you have to consider by energy usage, state, placement.

In this paper, we studied on a group of an unmanned aerial vehicle called quadrotor that has four fixed rotors. They are connected and the controller via a redundant network that keeps the whole system always in connection if one or more UAVs are unresponsive or down. None of the UAV has GPS, their path or mission is defined by a human operator from a web page which can operate as Intra or Internet. Each UAV runs its computer vision, flight dynamics, and location finding, and path tracking algorithms on board.

2. Material and Method

Unmanned aerial vehicles can also classify according to their flying characteristics; fixed-wing, rotorcraft (a rotary wing), blimps, and flapping wing. We can describe fixed-wing UAV as airplanes. This UAV needs a runway to takeoff and landing or sometimes a catapult to give necessary initial relative velocity. They have long durability and cruise at high speeds. Because of these properties, they generally used for military applications.

Rotorcraft UAV is also called takeoff and landing (VTOL) or rotary-wing unmanned aerial vehicle vertically. Their main advantage is hovering and high maneuverability. These properties are the key features for many robotic missions and civilian applications. They also have different rotor configurations such as a single rotor (conventional helicopters), axial, coaxial rotor (Sikorsky X2), tandem (Chinook), tilt (V-22), intermeshing (Kaman K-MAX) and multi-rotor (quadcopter). [11]. We show the comparison of rotorcraft and fixed-wing in Table 1. Scores in Table 1 represent our opinions, which were concluded according to the related studies in literature and our prior experience in the field. The purpose of showing our thoughts as scores is to make it clear how we decided to use a quadcopter design for our research.

Fixed-wing aerial vehicles have high cargo (payload) capacity with high airspeed, and they can be constructed in a smaller size but with high aerodynamically stable construction. Despite all these properties, this design is lack of stationary flight. And it also has the lowest maneuverability; fixed wings cannot merely turn both sides without drawing an arc in the air. A single rotorcraft vehicle is also known as the helicopter has a proven design with high maneuver capacity. But it has an intricate mechanical and aerodynamic design. Control of a helicopter is also complicated. All the flight system equipment is critical during flight, and only a redundant structure can be the main motor. Axial rotors are very simple and construable with less power usage. But they are not for carrying a load, surveillance, and like other missions. Co-axial design is an improved design of a helicopter which is using two contra-rotating propellers. This aerial vehicle does not need any tail rotor. So, controllability and survivability property is higher than the conventional helicopter. Tandem rotor design is similar to co-axial rotorcraft in many aspects. Tiltrotor aerial vehicles are a relatively new concept. They are the combination of airplane and tandem rotorcraft.

Table 1. Comparison table of aerial vehicles (Higher is better, cross-relationships were neglected.)

Vehicle Property	Fixed Wing	Single	Axial	Co-Axial	Tandem	Tilt	Intermeshing	Multi
Flight Time (Power usage)	4	3	5	2	2	2	2	1
Controllability	4	1	1	3	2	4	4	5
Payload	5	3	1	4	4	4	5	3
Maneuverability	1	4	1	2	3	3	2	3
Mechanic Simplicity	4	1	5	3	2	1	2	5
Aerodynamic Simplicity	2	1	1	1	1	2	1	3
Flight Speed	5	4	1	3	2	5	3	4
Miniaturization	5	4	5	4	3	1	2	4
Survivability	4	1	1	3	2	5	2	3
Stationary flight	0	3	2	4	3	3	3	5
TOTAL	**34**	**25**	**23**	**29**	**24**	**30**	**26**	**36**

They are a higher survival rate than all aerial vehicle designs. Flight speed can be high as fixed wings. Intermeshing rotorcraft design, also known as synchropter, is a complex design of co-axial design. Two propellers have a certain tilt angle, and with this design concept tail rotor is unneeded. This design advantage is to carry about 2-3 tons of payloads. Multirotor design has advantages of the simpler controller, mechanic, and aerodynamic design. With the help of the number of rotors, stability is high. But the main disadvantage of this design is lower flight time with high power consumption.

The quadcopter can command only four-rotor speeds. But it has six degrees of freedom, three rotational and three translational. These motions are coupled to perform 6 DoF. Therefore, the dynamics of the quadcopter are nonlinear, especially when aerodynamics effects are considered. Aircraft have to supply their damping for stopping and hovering because aircraft have very less amount of friction than ground vehicles.

The UAVs that are used in this study are AR Drone 2.0, Parrot Inc. We have 6 of them in our experiments to simulate multiple robot control. We customize one of them for automatic charging. One of them is left original for backup. If any defect occurs in custom kernel compilation, we use it for recovery.

2.1. Hardware specification

AR Drone is a low-cost, high, highly skilled × type quadcopter from Parrot, as shown in Figure 1. It has a water-resistant body with a robust structure via carbon fiber tubes. It is 420g with the indoor hull. The quadcopter has four 15Watt brushless in runner motors. Motor's maximum rotation speed is 41400RPM. Normal speed during flight is about 28000RPM. 8" propellers with high attack angle are driven via 1/8.75 reduction from motors axes for high torque gains. Quadcopter's power source is a single Li-Po battery, which is 1000mAh with a 10C discharge rate. AR Drone also has a fully open-sourced software developer kit (SDK) for academic and personal usage. The main control board has a 1GHz 32-bit ARM Cortex A8 processor with 1GB 200 MHz DDR2 RAM. It also has an 800MHz dedicated video digital signal processor for the onboard vision system. The controller board operates with BusyBox embedded Linux distribution. And also, the central controller board has a USB 2.0 extension for extending connectivity. Communication with teleoperators or centralized task controller can be performed via the IEEE 802.11 wireless communication module.

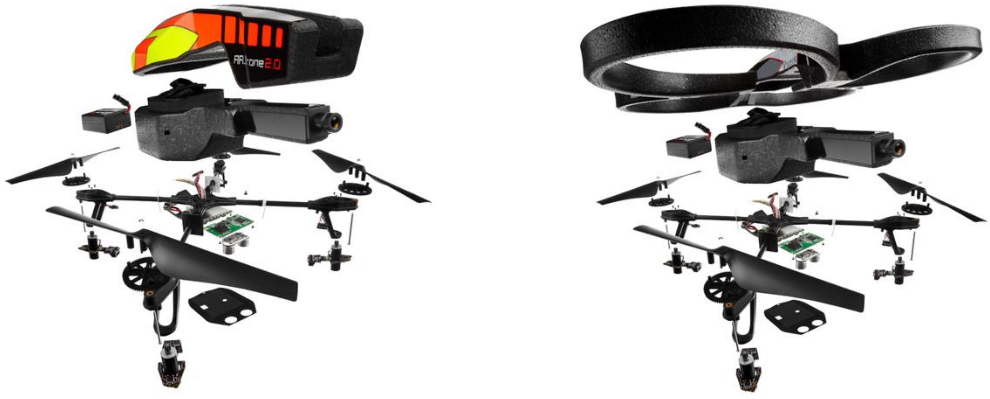


Figure 1. AR Drone quadcopter disassembled view

The quadcopter has a well-optimized internal measurement unit (IMU). In this IMU, three-axis gyroscopes with 2000 °/sec precision, three-axis accelerometers with ±50mg precision, three-axis magnetometers with 6° precision, pressure sensor ± 10Pa precision (80 cm at sea level) and an ultrasonic sensor for altitude control. AR Drone quadcopter also has two onboard cameras. One camera is on the front, and the other is at the bottom of the vehicle. The front camera has 1280x720 resolutions at 30 FPS. It has a wide-angle lens with a 92° field of view (FOV). Camera capture data is sent through the wireless link using H264 encoding for low latency streaming. The bottom camera has 320x240 resolutions at 60 FPS, and it also uses this camera for ground speed measurement [12].

2.2. Network design

Quadrotor has two main network design criteria. First one is to create a router with DHCP capability if there is no control network in the range. We control this and validate by querying SSID value in the current environment. Second one, if there is already our network is available at the range, join this network. With these features and a background daemon on the quadrotor's embedded system, we can manage communication between the computer and the UAVs with no interruption. Even in the worst-case scenario, when the main quadrotor that constructs the wireless network fails because of an empty battery or an instant crash, the background worker (daemon) of any other quadrotor can start the same wireless network. At that instant, the UAVs' missions can continue because of TCP/IP structure. If quadrotors already receive valid TCP packages, the task will continue to execute. In this scenario, there is no need for a centralized computing or controlling structure. All the maneuvers can be done using the onboard controller. The network controller algorithm concept expresses in Algorithm 1. pseudo-code block. This controller consumes CPU load in each drone, which is around %7. But with the help of it, we can keep the network stable and continuous. On the client-side, a web application is a daemon itself. It tracks and adjusts network changes in the background with all other functions so that the CPU load effect is negligible.

2.3. Control framework architecture

In this study, our main adjective is to construct a stable and robust web controller to use it in various scenarios. Frst step for stability is the network between the controller client and drones itself, which already defined in the previous section. Our web-based control framework can define as a data-intensive real-time application (DIRT). And it has some key design parameters. These parameters are portability, lightness, accessibility, extensibility.

Algorithm 1. Network controller daemon both for each drone and client

```
shared_const gateway, subnet
const static_ip forEach drones
const static_ip forEach clients
async.forEach drones
  if this.drone.ip is not router.ip and router.ip alive
    release router.ip
    release access_point
    join access_point
  if this.drone.ip is not router.ip and router.ip not alive
    create access_point
    publish new this.drone.ip as router.ip
async.forEach clients
  if router.ip is not alive
    release router.ip
    release access_point
    join access_point
```

To achieve these parameters, Node.js selected as the main runtime environment. It is a well-known open-source JavaScript runtime built on Google Chrome's V8 JavaScript engine in Figure 2.

Designed and developed framework can deploy any computer which is based on x86, x64, or ARM CPU architecture. For this study and A/B testing purposes, we had built it from the source under AR Drone's BusyBox shell. The split test shows us not any significant advantages. On the drone, we can use both factory setting configuration, alter these configurations, or use standard API to perform a stable flight, which we already achieved in our framework and study.

Node.js runtime itself is lightweight than other programming runtime machines such as Java Virtual Machine (JVM), Ruby on Rails (RoR), and Microsoft .NET.

This controller framework, thanks to Node.js's capability, can gain device-level access so that we can control both client's hardware such as a network card to maintain the connection. We use runtime's non-blocking I/O model to access and propagate mission-related commands to the whole drone network.

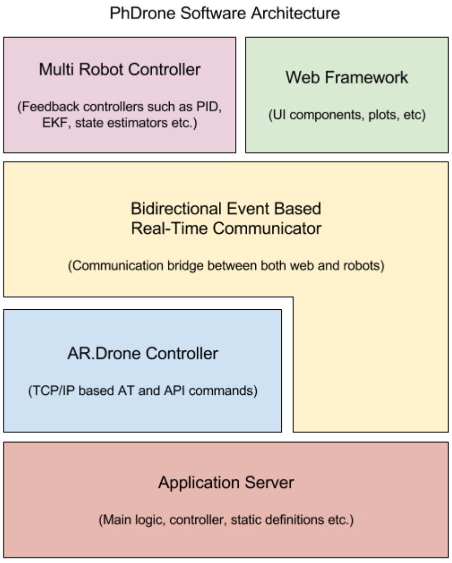

PhDrone Software Architecture
Multi Robot Controller
(Feedback controllers such as PID, EKF, state estimators etc.)
Web Framework
(UI components, plots, etc)
Bidirectional Event Based Real-Time Communicator
(Communication bridge between both web and robots)
AR.Drone Controller
(TCP/IP based AT and API commands)
Application Server
(Main logic, controller, static definitions etc.)

Figure 2. “PhDrone” Software Architecture

This asynchronous behavior can be defined as a failure tolerant server. This kind of server does not wait for an API to return any data; it just passes the next one. But also, to get tracks of previous API calls, we can use events. Every call can be tracked and monitored with events such as begin, ongoing, done, fail, or similar. Every call in the runtime environment uses a single thread. This call can observe as a disadvantage, but thread-based solutions in a complex real-time system is already a big challenge and had many disadvantages. Events can be used to overcome this bottleneck. Because the event mechanism helps the main loop to respond to any request in a non-blocking way. This feature creates high scalability to a backend server. In our study, this server controls every flight for each drone with a given lattice-based XY path and additionally altitude.

Control framework front-end can be accessed through client PC's IP address and pre-defined distinct port number, for example, 8000. Web GUI has all the features to assign path following missions, altitude steps for all drones in the “PhDrone" network. If a joint mission assignment is needed, triggering steps can also be defined. With the help of this kind of mission, the motion of one drone can guide another drone. This kind of mission scenario can be used for cascade group maneuvers.

3. Results

In this period of study, we can make our network connection much more stable and robust. Our web-based application enables all features which are defined in AR Drone's SDK. And also, with the help of an asynchronous web application, we can easily control drones in Figure 3.

During the experiments, we realized that optical flow heavily depends on background and light conditions like all other image processing techniques. But with the help of minor adjustments to PID control and Kalman filter, we get all these results without using any additional sensors or conditioned environment.

All the experiments are controlled, run, and report by a web application that is developed in this study. All paths for each drone can be assigned and preview on the web page. The lattice path approach generates paths. A lattice path is a walk in a lattice in some d-dimensional Euclidean space [1]. Formally, a lattice path P is a sequence P = (P0, P1, ..., Pn) of points Pn in Zd. The point P0 is called the starting point, and P is called the endpoint of P. The vectors P0, P1, P2, ..., Pn−1, Pn are called the steps of P.

To keep formation with using the lattice path, we construct a world and body frame line in Figure 4. All measurements are referred to as the world frame (W), corresponding to axes Wx, Wy, and Wz. The origin of the formation body frame (B) that have axes Bx, By and Bz. The body frame was chosen to coincide with the centroid of the triangle formation shape.

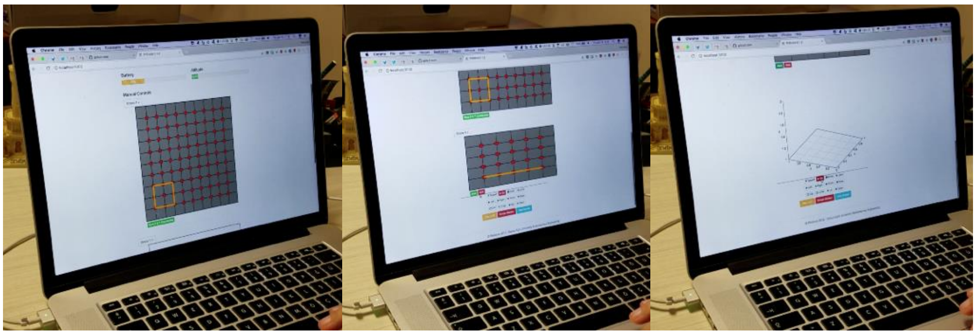


Figure 3. Doing experiments through the web application

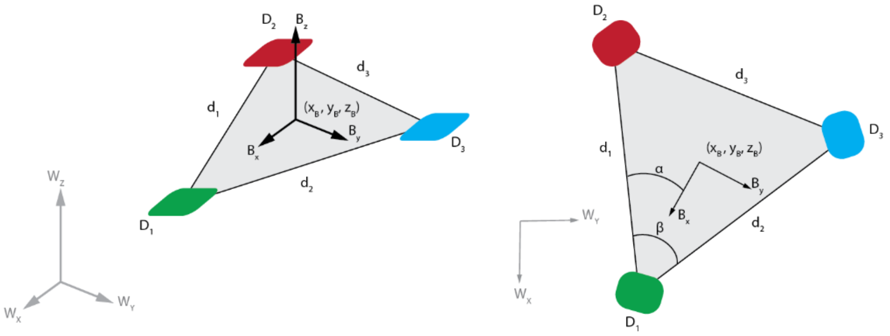


Figure 4. Perspective and top view of the triangle formation, side by side

The centroid of formation triangle, P_B, and shape matrix S_B constructs with the following equations.

$$P_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} \frac{x_1+x_2+x_3}{3} \\ \frac{y_1+y_2+y_3}{3} \\ \frac{z_1+z_2+z_3}{3} \end{bmatrix} \qquad \text{Eq. 1}$$

$$S_B = \begin{bmatrix} d_1 \\ d_2 \\ \beta_B \end{bmatrix} = \begin{bmatrix} \sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2} \\ \sqrt{(x_1-x_3)^2+(y_1-y_3)^2+(z_1-z_3)^2} \\ \cos^{-1}\frac{d_1^2+d_2^2-d_3^2}{2d_1\, d_2} \end{bmatrix} \qquad \text{Eq. 2}$$

$$d_3 = \sqrt{(x_1-x_3)^2+(y_1-y_3)^2+(z_1-z_3)^2} \qquad \text{Eq. 3}$$

$$B_x = D_1 - P_B = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} - \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \qquad \text{Eq. 4}$$

$$B_z = (D_1 - D_2) \times (D_1 - D_3) \qquad \text{Eq. 5}$$

$$B_y = B_x \times B_z \qquad \text{Eq. 6}$$

$$Roll\ \phi_B = \tan^{-1}\frac{B_{x(y)}}{B_{y(x)}}\ , Pitch\ \theta_B = \sin^{-1} B_{x(z)}\ , Yaw\ \psi_B = \tan^{-1}\frac{B_{y(z)}}{B_{z(z)}} \qquad \text{Eq. 7}$$

In experiment # 5, we drive 2 (two) drones, which have 20cm distance at the resting point. Although their airflow interfered with each other in Figure 5, the mission is successful.

All of first 5 experiment flights done in a closed office room. The floor has wooden like covering, as shown in Figure 6. This overlay has an advantage for the computer vision algorithm. Because it is not reflective, and its color is homogeneous.

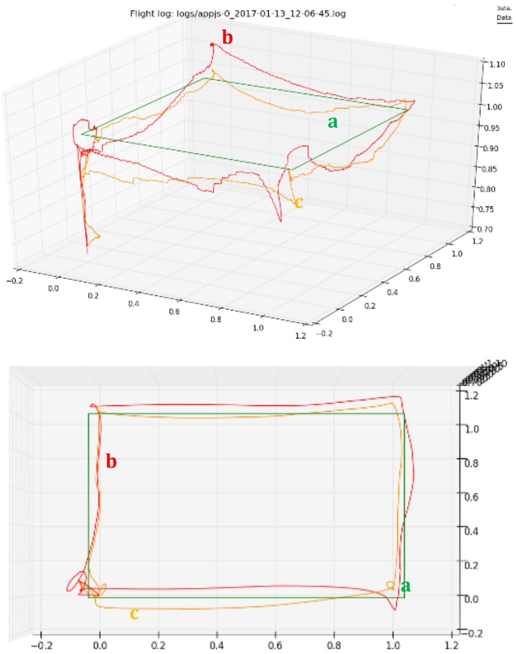


Figure 5. Experiment # 5 – 2 (two) drones are drawing a 1m x 1m square at 1m altitude with fixed heading (a-green; given path, b-red and c-orange; measured data)

But in our case, we are trying to control the flight of quadcopters in big, large indoor areas such as airports or factories. After the fifth experiment (there are more but not included to maintain the integrity of this report), all other experiments were placed in a factory warehouse, which has a tough concrete floor.

First experimental mission path at the warehouse is to follow a right triangle, as shown in Figure 7. The lattice map defined as following [13, 14]:

$$Map_{P4b} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

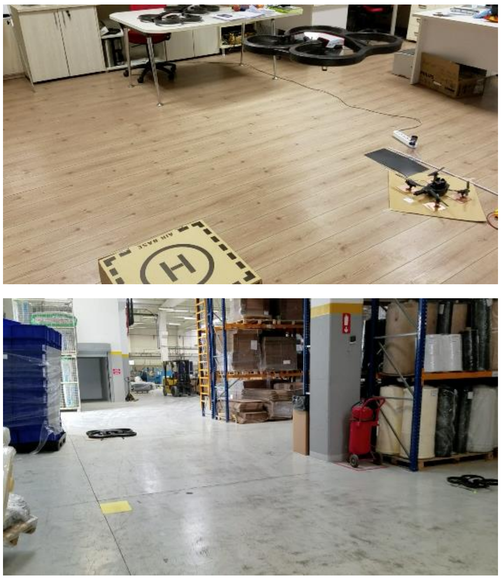


Figure 6. Office floor (up) vs. Warehouse floor (down)

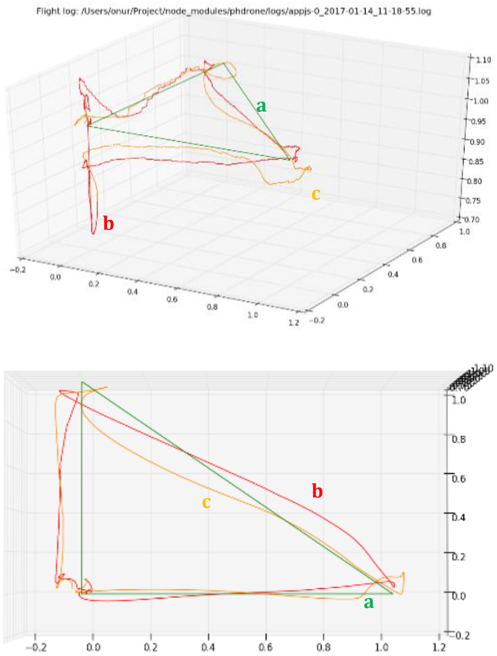


Figure 7. Experiment # 6 Single drone in sequential flight, right triangle-shaped path mission at a warehouse (a-green; given path, b-red and c-orange; measured data)

In experiment #6, one drone flights one more time, right after first landing, to experience mission consistency in a different area in Figure 6.

Another triangle-shaped mission path is also studied. The lattice map defined as follow:

$$Map_{P4b}\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}0&0&1&0\\0&1&0.5&0\\1&1&1&1\end{bmatrix}$$

Mission as shown in Figure 8. like experiment # 6, a single drone is used and gather sequential data. This experiment shows that we still have larger errors while making cross movements. But, if the floor is not reflective and has no pattern, the optical flow algorithm can correct the path during flight. This correction can be seen from P$_1$ (0.5, 1.5) to P$_2$ (1.5, 2.0) in Figure 8.

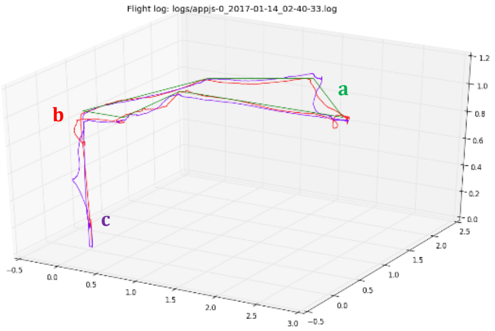


Figure 8. Experiment # 10 "mountain" like a path at 1m altitude with fixed heading (a-green; given path, b-red and c-purple; measured data)

After getting results like experiment # 12, we want to focus cross movements and see in a basic map, which can be seen in Figure 9. But when we look at R2 of the mission path, we realized that we almost approach 1.0, our real R2 value is 0.99277, as seen in Figure 9. But in real-time systems mainly aerial control systems, the frequency of the position and displacement sensor has to be more than 60Hz. In our case, our drone's vertical camera can get 60Hz frames, but processing it on-board has some limitations.

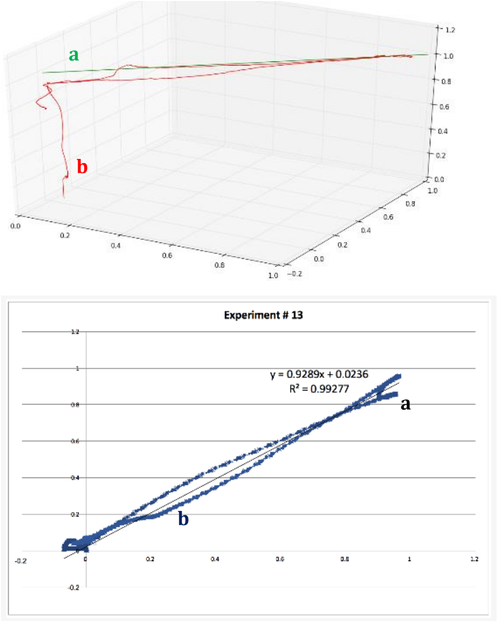


Figure 9. Experiment # 13 "slash" like path to examine position errors in cross-movement (top – c-green; given path, b-red; measured data, bottom – a-black; given path, b-blue; actual data) and simple linear regression and R2 calculation for basic “slash” like mission (bottom)

For a group flight experiment, the assigned task is to perform a sequential flight in a triangular shape in Figure 10. Quadrotors are starting the mission on each of the edges; when the ith quadrotor moves ith + 1 position, ith +2 quadrotor moves ith position in Figure 11. The task is just like a puss-in-the corner game or like vultures draw circles in the air. In this experiment, we show that a quadrotor can trigger another quadrotor when it finishes its job. As seen in Figure 12, the mean error of the x and y-axis is between 9 to 20 cm. For a GPS-denied, vision, and essential IMU-based flying platforms, this error can be acceptable when considering GPS accuracy is about 4.6 m [15].

Flight log: logs/201706\multi-192.168.1.102_1497574519397.log

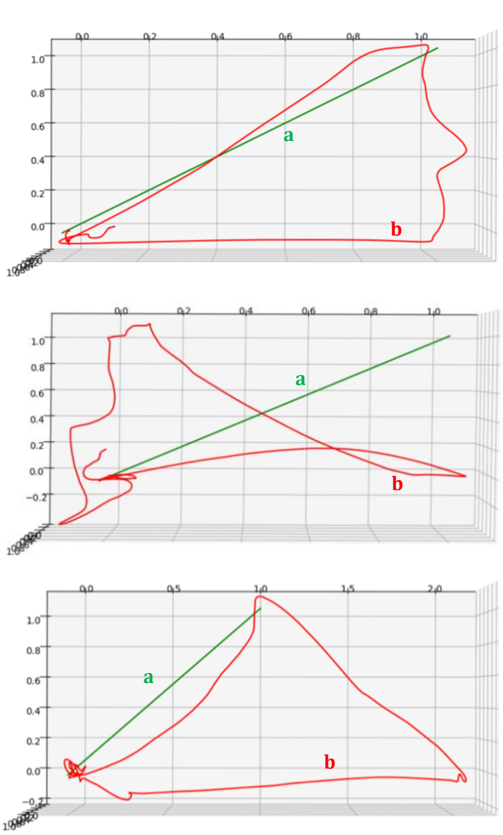


Figure 11. Actual flight data from a group flight (c-green; given path, b-red; measured data)

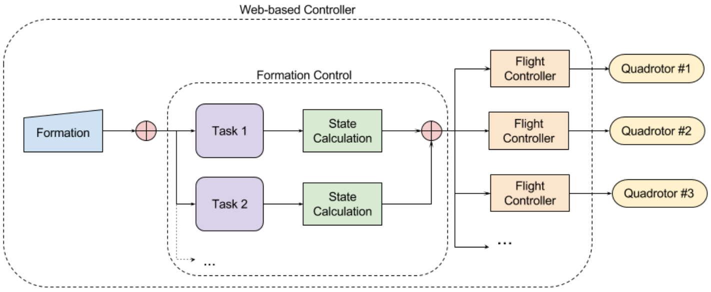
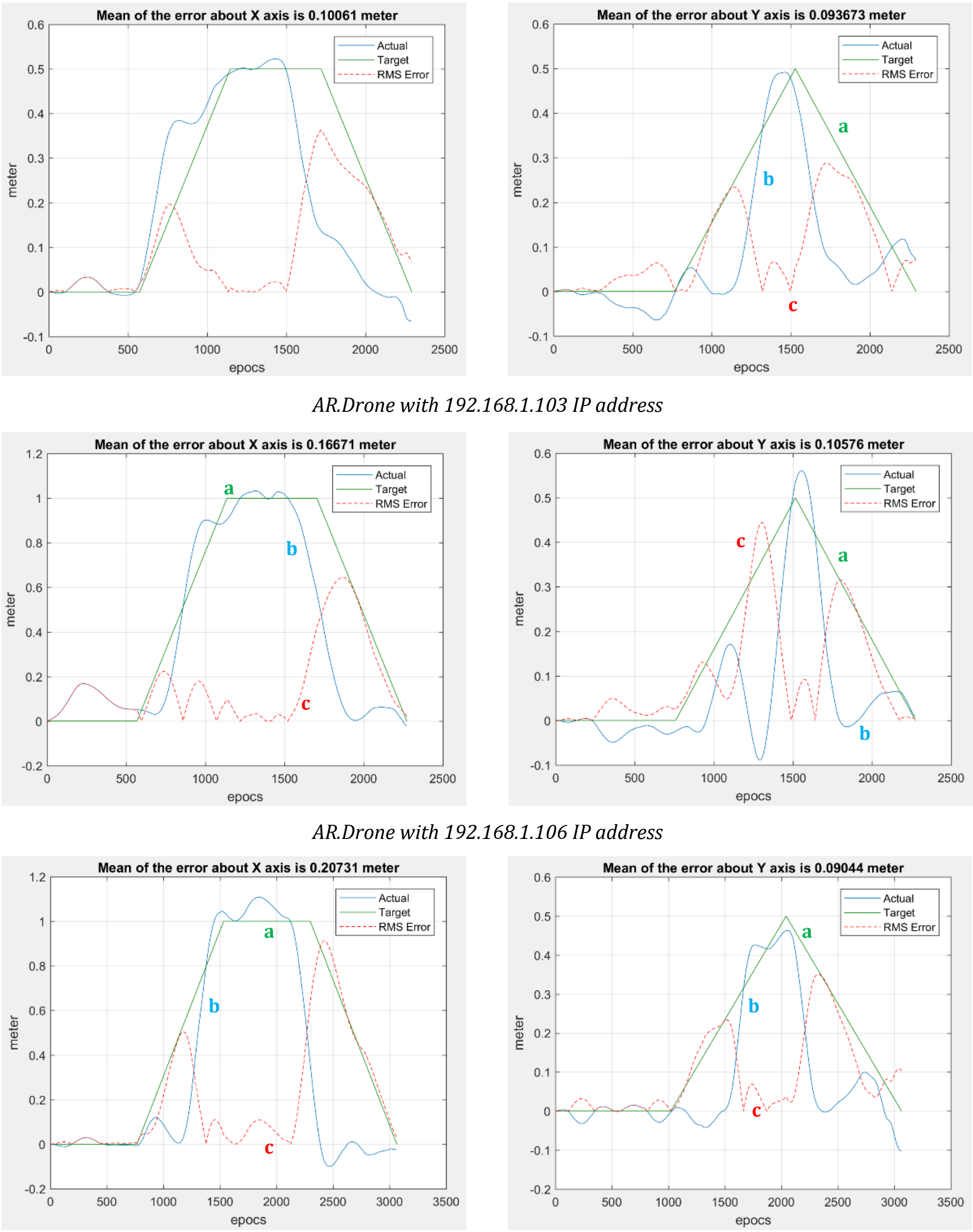


Figure 10. The control loop of a group flight

AR.Drone with 192.168.1.102 IP address



AR.Drone with 192.168.1.103 IP address

AR.Drone with 192.168.1.106 IP address

Figure 12. Group flight experiments' accuracy analysis of group flight (a-green; target, b-blue; actual, c-dashed red; RMS error)

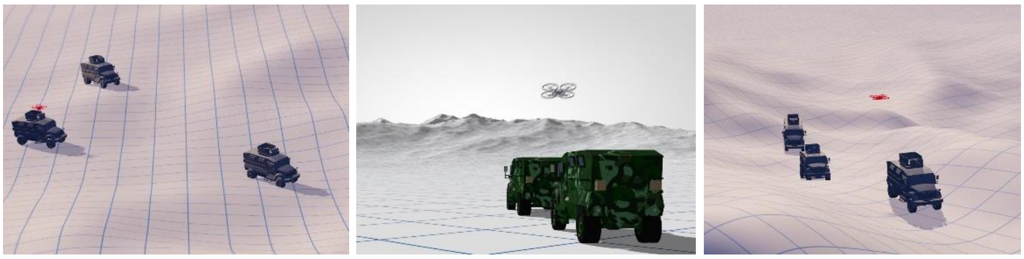


Figure 13. Simulation environment screenshots of the military-specific missions

4. Discussion and Conclusion

In this study, we successfully control multiple drones via a TCP/IP network by constructing a self-healing network structure. The controller framework runs under a regular web page with a non-blocking event structure. We can easily manage the health status of all the networks, and if something is wrong, they can warn users on the front-end, and reconnection attempted to establish as soon as possible.

All the control routines publish through TCP/IP protocol. During any interruption occurs in the data or mission transmission, the part that successfully transmits executed by the relevant drone. Each drone is using its own IMU and vision systems that are already on board to execute given path missions. But we calculate the tracked orientation and translation in each step. With the help of these calculations, our framework can determine each drone's current state. We can use this feedback mechanism in GPS-denied or indoor environments with a tolerance of +/- 500mm. There is no global visioning system for multiple or a single UAV. There may be some predefined and known landmarks from AR or QR codes, passive RFID tags, or even Wi-Fi routers/antennas for enhancing the accuracy of positioning and calculation or correction of the distance over triangulation methods.

After successfully controlling a group of quadrotors in an indoor environment, applying the same control and command principles, a single quadrotor mission performed for military-specific missions like in Figure 13. In that mission, the quadrotor was successfully taken off from a vehicle in motion with 50 km/hr with a 45o of departure angle from its automatic gripping nest. After executing the given lattice-path mission for a radius of 10-meter, the quadrotor can land its nest while the vehicle is on hold. For further works, the landing procedure will be automatized. At the same time, the vehicle is also on the move, and multiple UAV flight traffic above land vehicles will be studied to assign discrete and joint missions.

Acknowledgment

This study is a part of the 2012.KB.FEN.116 project that is supported by Scientific Research Projects Coordination Unit of Dokuz Eylül University.

References

[1] Bachrach A. G. 2009. Autonomous flight in unstructured and unknown indoor environments. Ph.D. dissertation, Massachusetts Institute of Technology

[2] Bouabdallah S. 2007. Design and control of quadrotors with application to autonomous flying. Ph.D. dissertation, Ecole Polytech Federale de Lausanne, Lausanne, Switzerland

[3] Bry A., Bachrach A., and Roy N. 2012. State estimation for aggressive flight in GPS-denied environments using onboard sensing. Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, pp. 1–8

[4] Bills C., Chen J., and Saxena A. 2011. Autonomous mav flight in indoor environments using single image perspective cues. Robotics and Automation (ICRA), 2011 IEEE international conference on. IEEE, pp. 5776–5783

[5] Roberts J. F., Stirling T., Zufferey J.-C., and Floreano D. 2007. Quadrotor using minimal sensing for autonomous indoor flight. European Micro Air Vehicle Conference and Flight Competition (EMAV2007), no. LIS-CONF-2007-006

[6] Valenti M., Bethke B., Fiore G., and How J. 2006. Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery. in Proc. AIAA Guidance, Navigation, and Control Conf. and Exhibit, Keystone, CO, Aug., pp. 2006–6200

[7] Venugopalan, T. K., Taher, T., & Barbastathis, G. 2012. Autonomous landing of an Unmanned Aerial Vehicle on an autonomous marine vehicle. In Oceans, 2012 (pp. 1-9). IEEE.

[8] Richards A., Bellingham J., Tillerson M., and How J. 2002. Co-ordination and control of multiple UAVs. Proc. AIAA Guidance, Navigation, and Control Conf. and Exhibit, Monterey, CA, Aug., pp. 2002–4588

[9] Vago Santana, L., Brandao, A. S., Sarcinelli-Filho, M., & Carelli, R. 2014. A trajectory tracking and 3d positioning controller for the ar. drone quadrotor. In

Unmanned Aircraft Systems (ICUAS), 2014 International Conference on (pp. 756-767). IEEE.

[10] Hoffmann G., Waslander S., and Tomlin C. 2008. Quadrotor helicopter trajectory tracking control. Proc. AIAA Guidance, Navigation, and Control Conf. and Exhibit, Honolulu, HI, Apr., pp. 2008–7410

[11] Watkinson J. 2004. Art of the Helicopter, 9780080472034

[12] Krajnık T., Vonasek V., Fiser D., and Faigl J. 2011. AR-drone as a platform for robotics research and education. In Proc. of the Communications in Computer and Information Science (CCIS)

[13] Krattenthaler C. 2015. Lattice Path Enumeration. Handbook of Enumerative Combinatorics, M. Bona (ed.), Discrete Math. and Its Appl., CRC Press, Boca Raton-London-New York, pp. 589-678

[14] Wallner M. (2016) Lattice Path Combinatorics. Ph.D. dissertation, Technischen Universitat Wien

[15] Diggelen, F. and Enge, P. 2015. The World's first GPS MOOC and Worldwide Laboratory using Smartphones. Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015), 361 – 369