UAV Delivery Monitoring System

Khin Thida San[1], Sun Ju Mun[1], Yeong Hun Choe[1] and Yoon Seok Chang[1]

*[1]UTAC, School of Air Transportation and Logistics, Korea Aerospace University,
Goyang City, Seoul 412-791, Republic of Korea (South Korea)*

Abstract. UAV-based delivery systems are increasingly being used in the logistics field, particularly to achieve faster last-mile delivery. This study develops a UAV delivery system that manages delivery order assignments, autonomous flight operation, real time control for UAV flights, and delivery status tracking. To manage the delivery item assignments, we apply the concurrent scheduler approach with a genetic algorithm. The present paper describes real time flight data based on a micro air vehicle communication protocol (MAVLink). It also presents the detailed hardware components used for the field tests. Finally, we provide UAV component analysis to choose the suitable components for delivery in terms of battery capacity, flight time, payload weight and motor thrust ratio.

1 Introduction

The use of UAVs in urban areas is increasingly being considered to overcome ground delivery traffic issues. However, flight environments are subject to dynamic changes due to shifts in weather patterns and moving objects. To improve UAV flight safety, proposals have been offered to effectively monitor the entire UAV delivery process. This paper provides requirements that should be considered when building an effective ground control station (GCS). There are two main GCS types – ready-made open source GCS and customized GCS.

Ready-made GCS is typically open source software that can easily be downloaded from the internet and used for mission management and real-time flight operation. Lim et al. [1] analyzed eight open-source projects (OSPs) for UAVs. They discussed five autopilot components – processors, gyroscopes, accelerometers, magnetometers, and barometers – and introduced flight controllers with various functions to monitor and control UAVs. Moreover, additional functions can be added by extending the open source code. Mission Planner and Qgroundcontrol are representative open source software packages that deliver UAV flight automation. It enables UAVs to fly along predetermined waypoints and provides real time flight data. It also provides an automatic return function that engages when fuel or battery levels are low [2]. Ready-made GCS are convenient due to their wide functionality range. However, some functions of logistics purpose for delivery tracking in real time and history report of the previous flight missions are not provided. And, they are overly focused on flight functions, making it too difficult for normal users to apply for non-traditional tasks.

Customized GCS can be implemented according to project requirements. For example, Perez et al. [3] described the development of an efficient and comprehensive guidance system to monitor multiple UAVs in real time, while also discussing the main layout for a multi-UAV GCS. Generally, open source libraries are used to develop customized GCS. Dong et al. [4] developed a real-time software system for an unmanned helicopter with a two-layer framework, i.e. the data was transferred in the background while the visualization operated in the foreground.

The proposed monitoring system is a customized GCS inspired by FlightRadar24 (Live Air Traffic) [5]. FlightRadar24 facilitates the real time tracking of commercial flights by providing geographical locations and flight data such as flight numbers, speeds and headings. The monitoring system provides not only flight data but also delivery item management to improve logistics efficiency.

In this research, Pixhawk autopilot UAV is used with selected flight components to perform deliveries. MAVLink protocol [6-8] is used to receive the flight data and command of the UAV. Our main objectives for the monitoring system are (1) managing order assignment for a flight mission (item assignment and path planning), (2) controlling autonomous flight, (3) monitoring in real time, (4) tracking the UAV and delivery items, and finally, (5) dropping the parcel at a specific location autonomously (gripper servo movement).

2 System structure

Fig.1 depicts the required functions of the monitoring system for UAV delivery. The system is designed to combine useful UAV delivery system functions that are not normally supported by ready-made GCS.

https://doi.org/10.1051/matecconf/201815104011

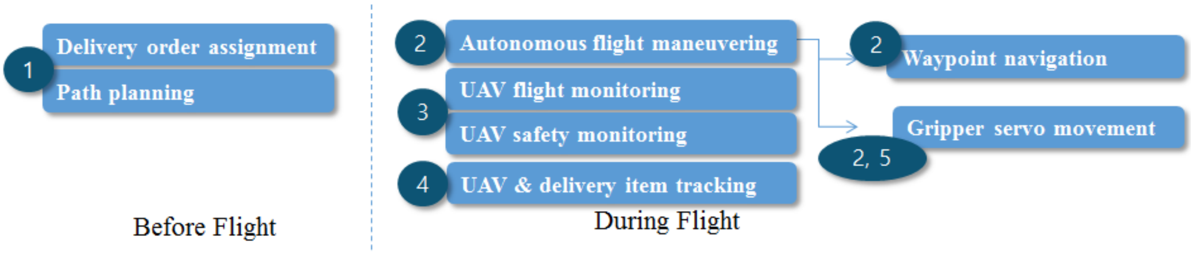


Figure 1. UAV delivery monitoring system functions

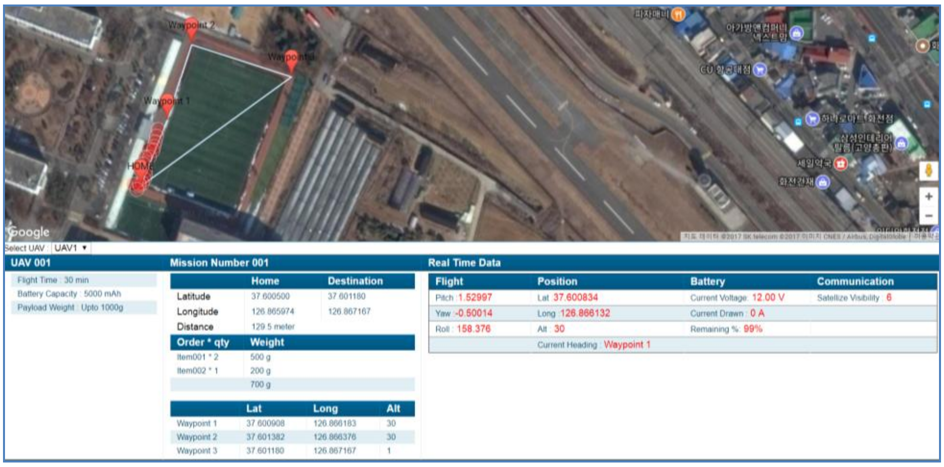
(1) The system combines order assignment and path planning functions together to deliver-items with minimal flight times. We use a concurrent scheduler approach together with a genetic algorithm (GA) to assign items to flight missions [9-10]. Accordingly, there are six parameters – item weight, order weight (for a collection of items), the UAV's sustainable payload, UAV flight time, delivery distance and UAV speed. All of the parameters must satisfy one another to achieve real flight. Therefore, GA with the concurrent scheduler approach is used to solve multi-parameters. Finally, the approach results in sets of missions with items, destinations, and selected UAV.

(2) The autonomous flight is controlled via MAVLink. The commands with detailed parameters for the required maneuvers are shown in Table 2.

(3) For real time monitoring and UAV flight safety, our monitoring system uses MAVLink messages to receive autopilot flight data, which are shown in Table 1.

(4) UAV and item tracking are shown in Fig. 2. Three main panels – UAV details, mission details, and real time data – are included. The UAV details panel shows detailed data regarding UAV flight time and sustainable payload weight. The mission details panel shows the current mission and ordered items information, as well as the home, destination, and waypoints. The real time data panel shows the real time data for the flights data, positions data, battery data, and telecommunication data.

(5) Finally, the specific destinations (latitude and longitude) are defined as waypoints to drop parcel. The system tracks UAV location during flight. UAV receives a command (MAV_CMD_DO_SET_SERVO [11-12]) to open the gripper servo when it reaches the proper altitude of the designated destination (MAVLink command, Table 2, No 6). The whole UAV delivery procedures can be seen in Fig. 3. The next steps of the delivery system are to implement emergency flight control with the help of the environment data and UAV components failure monitoring.



UAV 001

Flight Time : 30 min
Battery Capacity : 5000 mAh
Payload Weight : Upto 1000g

Mission Number 001

	Home	Destination
Latitude	37.600500	37.601180
Longitude	126.865974	126.867167
Distance	129.5 meter	

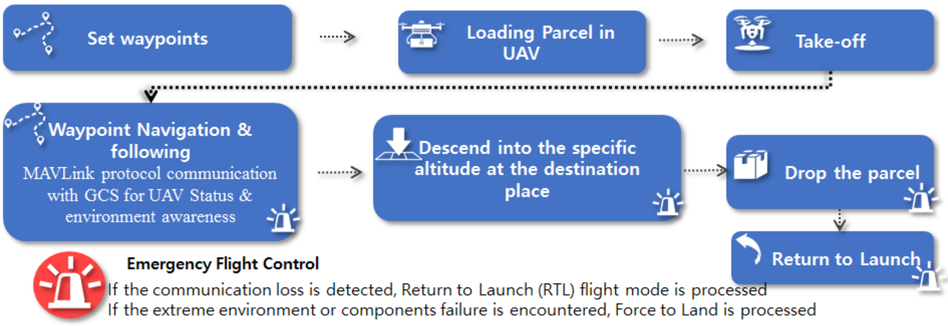
Order * qty	Weight
Item001 * 2	500 g
Item002 * 1	200 g
	700 g

	Lat	Long	Alt
Waypoint 1	37.600908	126.866183	30
Waypoint 2	37.601382	126.866376	30
Waypoint 3	37.601180	126.867167	1

Real Time Data

Flight	Position	Battery	Communication
Pitch : 1.52997	Lat : 37.600834	Current Voltage : 12.00 V	Satellize Visibility : 6
Yaw : -0.50014	Long : 126.866132	Current Drawn : 0 A	
Roll : 158.376	Alt : 30	Remaining % : 99%	
	Current Heading : Waypoint 1		

Figure 2. UAV delivery monitoring system



If the communication loss is detected, Return to Launch (RTL) flight mode is processed
If the extreme environment or components failure is encountered, Force to Land is processed

Figure 3. The whole UAV delivery step by step procedures

3 Communication structure

In the system, 3DR telemetry (TX/RX) is used to communicate between the Pixhawk autopilot UAV and monitoring system. Pixhawk supports PX4 firmware and Ardupilotmega firmware. We use Ardupilotmega for our system. Specific system IDs are assigned to each autopilot and they are assumed as the UAV ID in MAVLink. The telemetry conveys MAVLink messages and data which contain the aircraft flight data and UAV GPS information with the respective system ID of the XML data file. At the same time, the monitoring system sends commands to the UAV to engage in functions such as take-offs, landings, and auxiliary servo movements. The commands are also sent to UAV via the command XML file. XML files are generated from the MAVLink C header library, which is decoded or encoded using Node.JS MAVLink of the JavaScript package manager [13]. We use two telemetries – TX and RX. One of the telemetries (Serial COM6) is used to send while the other (Serial COM5) continues to receive telemetry messages, as shown in Fig. 4.

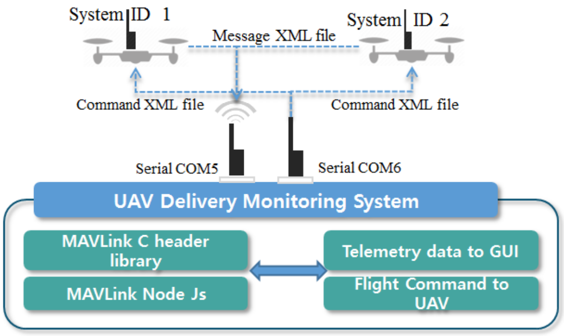


Figure 4. Communication for UAV delivery monitoring system

4 MAVLink communication protocol commands and messages

Tables 1 and 2 describe the MAVLink messages and commands used to implement the monitoring system. Table 3 describes the report status messages (failed or executed) that appear after the commands are sent.

Table 1. MAVLink Message

	Description/ Purpose/ Message ID	**Field name and Remark**
1	-Geographical location -UAV tracking GPS_RAW_INT [14]	lat (Latitude)
		long (Longitude)
		alt (Altitude)
2	-Satellite visibility -UAV communication GPS_RAW_INT [14]	satellites_visible (Number of satellites visible (0-20) – small visibility numbers indicate that communication between the UAV and GCS is weak) fix_type (status of GPS – No GPS, 3D GPS and 2D GPS)
3	-Current waypoint -UAV tracking MISSION_CURRENT [14]	seq (Number of sequence/waypoint to which the UAV is currently heading. We define sequence as a waypoint before actual flight)
4	-Battery monitoring data -UAV flight safety SYS_STATUS [14]	battery_voltage (currently used battery voltage in real time of flight)
		battery_current (drawn current in real time of flight)
		battery_remaining (remaining percentage in real time of flight)
5	-Flight data -UAV flight safety ATTITUDE [14]	pitch (pitch movement in real time of flight) yaw (yaw movement in real time of flight) roll (roll movement in real time of flight)
6	-Speed data -UAV flight safety GPS_RAW_INT[14]	vel (Ground speed - provided by GPS embedded on autopilot)
7	-Heading data -UAV flight safety GLOBAL_POSITION_INT [18] GPS_RAW_INT[14]	hdg (UAV heading yaw angle in degree) cog (Course over ground – direction of UAV movement between two waypoints with respect to the surface of the Earth)
8	-Flight data -UAV flight mode [15] HEARTBEAT [14]	base_mode custom_mode (The combination of base_mode and custom mode can change the flight mode details in Table 4)

Table 2. MAVLink Commands

	Description/Purpose/ Command ID	**Remark and parameters**
1	Defining waypoints for a mission/ flight To define waypoints for navigation MISSION_COUNT [14] MISSION_REQUEST [14] MISSION_ITEM [14] MAV_CMD_NAV_WAYPOINT [14]	To define waypoints into autopilot, a set of commands are used [16]. Sample waypoint assigning with MAVLink via Nodejs Javascript library can be seen in Fig. 5.

2	Mission clear
	To clear all waypoints of the autopilot
	MISSION_CLEAR_ALL [14]
		"MISSION_CLEAR_ALL",
		{
		'target_system': 1,
		// UAV ID
		'target_component': 1,
		// Autopilot
		'mission_type': 0
		//MAV_MISSION_TYPE = 0
		}.

3	Mode control
	For UAV flight mode
	SET_MODE[14]
		"SET_MODE",
		{
		'target_system' : 1,
		'base_mode' : 89,
		'custom_mode' : 3
		}
		(The combination of base_mode and custom mode can change the flight mode , detail in Table 4)

4	Take off / Landing To take-off and land COMMAND_LONG [14] MAV_CMD_NAV_TAKEOFF [14] MAV_CMD_NAV_LAND [14]

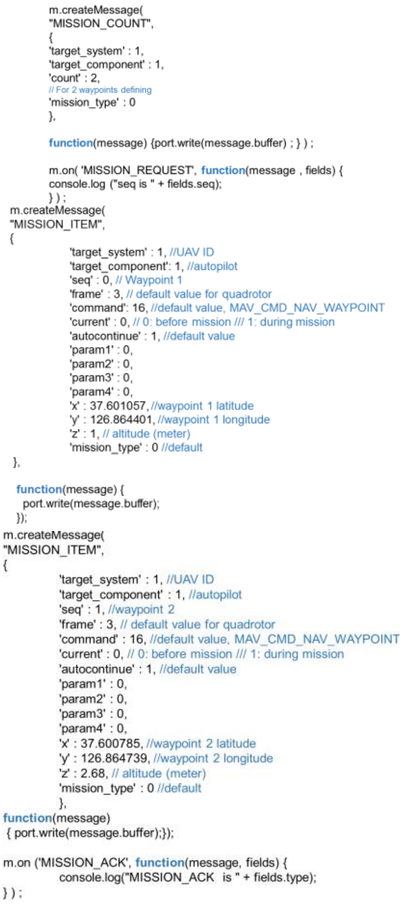
```
"COMMAND_LONG",
 {
  'target_system' : 1,
  'target_component' : 1,
  'command':22,
  //22 MAV_CMD_NAV_TAKEOFF
   //21 MAV_CMD_NAV_LAND
   'confirmation' : 0,
   'param1' : 0,
   'param2' : 0,
   'param3' : 0,
   'param4' : 2,
   'param5' : 37.600785,
   'param6' : 126.864739,
   'param7' : 2
 }
```

5	Servo control For the gripper servo movement in order to drop the parcel COMMAND_LONG[14] MAV_CMD_DO_SET_SERVO [12]

```
"COMMAND_LONG",
  {
   'target_system': 1,
   'target_component': 1,
    'command' : 183,
    //MAV_CMD_DO_SET_SERVO
    'confirmation' : 0,
    'param1' : 10,
    ///auxiliary 1 servo number
    'param2' : 1500,
    ///PMW 1000 servo is closed
    ///PMW 1500~1300 servo is opened
    'param3' : 0,
    'param4' : 0,
    'param5' : 0,
    'param6' : 0,
    'param7' : 0
  },
```

6	Arm/disarm To arm and disarm COMMAND_LONG MAV_CMD_COMPONENT_ARM_DISARM [14]

```
"COMMAND_LONG",
  {
   'target_system' : 1,
   'target_component' : 1,
   'command' : 400,
   //MAV_CMD_COMPONENT_ARM_DISARM
   'confirmation' : 0,
   'param1' : 1,
   ///1 arm //0 disarm
   'param2' : 0,
   'param3' : 0,
   'param4' : 0,
   'param5' : 0,
   'param6' : 0,
   'param7' : 0
  }
```



```
          m.createMessage(
          "MISSION_COUNT",
          {
          'target_system' : 1,
          'target_component' : 1,
          'count' : 2,
          // For 2 waypoints defining
          'mission_type' : 0
          },

          function(message) {port.write(message.buffer) ; } ) ;

          m.on( 'MISSION_REQUEST', function(message , fields) {
          console.log ("seq is " + fields.seq);
          } ) ;
     m.createMessage(
     "MISSION_ITEM",
     {
               'target_system' : 1, //UAV ID
               'target_component': 1, //autopilot
               'seq' : 0, // Waypoint 1
               'frame' : 3, // default value for quadrotor
               'command': 16, //default value, MAV_CMD_NAV_WAYPOINT
               'current' : 0, // 0: before mission /// 1: during mission
               'autocontinue' : 1, //default value
               'param1' : 0,
               'param2' : 0,
               'param3' : 0,
               'param4' : 0,
               'x' : 37.601057, //waypoint 1 latitude
               'y' : 126.864401, //waypoint 1 longitude
               'z' : 1, // altitude (meter)
               'mission_type' : 0 //default
      },

      function(message) {
       port.write(message.buffer);
      });
    m.createMessage(
    "MISSION_ITEM",
    {
              'target_system' : 1, //UAV ID
              'target_component' : 1, //autopilot
              'seq' : 1, //waypoint 2
              'frame' : 3, // default value for quadrotor
              'command' : 16, //default value, MAV_CMD_NAV_WAYPOINT
              'current' : 0, // 0: before mission /// 1: during mission
              'autocontinue' : 1, //default value
              'param1' : 0,
              'param2' : 0,
              'param3' : 0,
              'param4' : 0,
              'x' : 37.600785, //waypoint 2 latitude
              'y' : 126.864739, //waypoint 2 longitude
              'z' : 2.68, // altitude (meter)
              'mission_type' : 0 //default
              },
    function(message)
     { port.write(message.buffer);});

    m.on ('MISSION_ACK', function(message, fields) {
              console.log("MISSION_ACK  is " + fields.type);
    } ) ;
```

Figure 5. Assigning two waypoints with Node.JS MAVLink

Table 3. Acknowledgement message

	Description/Message ID	Field name and Remark
1	Command acknowledgement COMMAND_ACK [14]	result (MAV_RESULT) 0 - accepted 4 - failed This message is used after sending a command
2	Mission acknowledgement MISSION_ACK [14]	type (MAV_MISSION_RESULT) 0 - accepted 1 – failed This message is used after sending a series of sequences/ waypoints for a mission

Table 4. Flight Mode

Mode name	Base mode + Custom mode		Description
Stabilize	81	0	UAV can arm to fly only at the stabilize state
Alt Hold	89	3	UAV holds a specific altitude in order to drop the parcel from the air
RTL	89	6	Return to the start flying position after dropping the parcel
Auto	81	9	UAV changes into auto mode to navigate the predefined waypoints

5 Hardware analyses (flight time, weight and thrust force for motors)

For delivery purposes, the weight and flight time are correlated. Extra weight causes shorter flight times. Normally, flight time would be directly related to battery capacity. However, larger battery capacities may also result in heavier total payloads. Therefore, choosing an appropriate battery capacity often becomes a consideration. Normally, commercial UAVs can fly from 15 to 30 minutes. However, depending on customized components with advanced technologies and algorithms, some UAVs can fly for up to an hour.

5.1 Flight time estimation

Flight time estimation is not only effected by (i) the battery capacity but also (ii) the power consumption (current drawn) of the electronic components embedded in the UAV autopilot board. Moreover, for delivery purposes [17], (iii) the total all-up-weight (AUW), which includes the weight of products being delivered [17], and (iv) the thrust of the motor related to propeller size [18] are also considerable. A simple and an easy method of estimating flight time related to battery capacity and current draw is described in Eq.1.

$$\text{Flight time (minutes)} = (\text{Battery capacity}/\ \text{Current draw}) *60 \quad (1)$$

For example, if the battery capacity is 5600mAh (LiPoly 60C) and the motor amp has a current draw of 25, Eq. 1 produces a flight time estimation of 13.44 minutes. However, the flight time is around 11 minutes with around 900 grams of UAV weight together with 500 grams of carry payload in the real experiment [17]. Choosing the propeller size in order to lift the entire UAV mass is described in the load calculation equations and its related calculations [18].

5.2 Thrust motors to carry a specific payload weight (Thrust to weight ratio)

As flight time varies according to payload, the thrust required to move forward related to the total mass of the UAV should be calculated. Eq. 2 represents a commonly used, easy equation to calculate motor thrust. The final total AUW is the total weight of the UAV, including components, i.e. the airframe, autopilot, sensors, battery, motors, ESC, propeller, camera, and delivery package [18-20].

Required thrust per motor = [**AUW** x 2 (or) 3] / number of motors (2)

2:1 thrust ratio is standard where AUW is ratio value of 2 for one/each motor thrust. For particularly acrobatic cases that use a 3:1 ratio to improve efficiency. If a copter is able to handle approximately 1 kilogram, according to the 2:1 equation ratio, a total thrust of 2 kilograms and 500 grams of thrust per motor are required. In the present study, the thrust to weight ratio is set as 2:1; this is a good starting point as it ensures that enough thrust exists for the desired performance, and it avoids overly agile UAV movements.

5.3 Choosing the correct Lipo battery capacity

The electrical power for a multi-rotor can be approximated using Eq. 3 [21-23]. The equation is related to the power estimation theory of a Lipo electric battery (discharge rate).

Power (P) Watts = Voltage (V) V· Current (I) A (3)

The electrical power (watts) used by a multi-rotor to hover is P, calculated by obtaining V and I from the onboard power module. The motor speed and weight can vary according to the current draw under various conditions such as the UAV heading with air resistance. In addition, the battery discharge rate is higher in hot and warm weather than in cold conditions. Therefore, the maximum electrical power or maximum sustained discharge rate must be defined. In the example with 25 ˚C with 3700 mAh Lipo, the maximum sustained discharge rate is 92.5 A, which can be calculated as I = 25*3700/1000 = 92.5 A. For the quad-copter with four motors, the current draw for each motor is 21 A with a maximum (21x4) of 84 A. The amount of electricity needed for flight operation is 92.5 A (92.5 > 84). Accordingly, the maximum sustained discharge rate of 92.5 A with 3700 mAh is suitable for flight operation.

6 UAV Delivery Flight Experiment

The gripper for the parcel is attached under the UAV frame, as shown in Fig. 6 (left). Two servo motors are used to operate the gripper, Fig. 6 (right-up). The parcel box requires a stick with a head blob for the gripper to grab onto, Fig. 6 (right-down). Table 5 shows our UAV's component specifications for the parcel delivery.

Figure 6. Quad-copter with the parcel (left), the gripper with two servos (middle), and the parcel (right)

Table 5. UAV component specifications

	Component	Specification	Weight
1	Frame and landing skid	S500 glass fiber quad-copter frame motor centers: 480mm height: 170mm	405 g
2	ESC	Hobby Wing X Rotor 20A wired type	300 g
3	Propeller	Aerostar composite propeller/ 10x5.5cm	
4	Telemetry	3DR transceiver telemetry Radio 915mhz	
5	Autopilot	Pixhawk 2.1 standard set	
6	GPS	3DR GPS	
7	Motor	LDPOWER MT2213-920 KV/ Brushless motor Max Power 302.4W Max Amps: 18A	221.2 g
8	Battery	3DRobotics / Lipo Battery , 11.1V 8C, 5100mAh	320 g
9	Two servos (Gripper) and UBEC	HS-7955 TH HV Digital servo/ UBEC model HobbyWing 3A,MAX 5A 2-6S Lipo	220 g
Total weight			1466.2 g
Parcel weight (payload)			700 g
Total weight			2166.2 g (2.1 kg)

g = gram, kg = kilogram

A flight test is performed to accomplish the continuous three waypoints, as shown in Fig. 2. The UAV starts from the home location with an altitude of 30 meters and changed into auto mode to follow the waypoints. It then descends to 1 meter when it reaches Waypoint 3. The monitoring system subsequently detects the UAV at a specific location (latitude, longitude and altitude). It sends a servo command to open the gripper to drop the parcel. UAV flight mode then changes to return-to-launch (RTL). This returns the UAV to its home position. The entire mission lasted a little over 2.5 minutes with a distance of 129.5 meters from home to the destination. The delivery process was achieved autonomously from take-off to landing.

6. Conclusions

UAVs are capable of safely delivering goods via UAV monitoring functions. These functions monitor the working conditions of essential UAV components in real time, including batteries, speeds, headings, and flight data. In addition, this approach also provides tracking delivery statuses such as upcoming waypoints, current items, and descriptions. Moreover, the system can send immediate manoeuvring commands to a UAV to avoid obstacles and engage a forced landing. In the present study, our system is implemented to fulfil a fully autonomous UAV delivery to overcome last-mile logistics challenges.

Acknowledgment

This research was supported by Ministry of Science and ICT (MSIT), Korea, under Information Technology Research Center (ITRC) support program (IITP-2017-2014-0-00678) supervised by Institute for Information & communications Technology Promotion (IITP).

References

1. H. Lim,J. Park D. Lee, H.J.Kim, “Build Your Own Quadrotor: Open-Source Projects on Unmanned Aerial Vehicles,” *IEEE Robotics & Automation Magazine*, **19**,Issue. 3, 33-45 (2012).
2. A. Khalid, K. McFall, “Aerial Robotic Autonomous Patrol and Surveillance System,” *14th AIAA Aviation Technology, Integration, and Operations Conference*, 1-10 (2014).
3. D. Perez, I. Maza, F. Caballero, D. Scarlatti, E. Casado, A. Ollero, “A ground Control Station for a Multi-UAV Surveillance System,” *J Intelligent Robot System*, **69**, 119-130 (2013).
4. M. Dong, B.M. Chen, G. Cai, K. Peng, “Development of a Real-time Onboard and Ground Station Software System for a UAV Helicopter,” *Journal of Aerospace Computing, Information and Communication*, **4**, 933-955 (2007).
5. https://www.flightradar24.com
6. http://qgroundcontrol.org/mavlink/start
7. S,n.d Balasubramanian, Information on “MAVLink *Tutorial* for Absolute Dummies (Part –I),” http://ardupilot.org/dev/docs/mavlink-commands.html
8. http://mavlink.org/messages/common
9. K.T. San, E.Y. Lee, Y.S. Chang, “The delivery *assignment* solution for swarms of UAVs dealing with Multi-dimensional Chromosome representation of Genetic Algorithm,” *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, IEEE Annual (2016).
10. K.T. San, Y.S. Chang, “Concurrent Scheduler Approach with Genetic Algorithm for Item Delivery by using swarm of UAVs,” *The Proceedings of 21st International Symposium on Logistics (ISL 2016) Sustainable Transport and Supply Chain Innovation*, 510- 517 (2016).
11. D. Becker, N. Palumbo, et al. “HIGH ALTITUDE GLIDER,” Final Design Report.
12. http://mavlink.org/messages/common#MAV_CMD_DO_SET_SERVO
13. https://www.npmjs.com/package/mavlink
14. http://mavlink.org/messages/common
15. http://ardupilot.org/copter/docs/flight-modes.html
16. http://qgroundcontrol.org/mavlink/waypoint_protocol
17. D**.** Robert**,** “Construction and Testing of a Quadcopter,” A senior project presented to the Aerospace Engineering Department California Polytechnic State University, San Luis Obispo (2013).
18. Md. R Haque, M.Muhammad, D. Swarnaker, M. Arifuzzaman, “Autonomous Quadcopter for Product Home Delivery,” *International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, (2014).
19. https://www.grc.nasa.gov/www/k-12/airplane/fwrat.html
20. https://aviation.stackexchange.com/questions/8316/how-to-calculate-the-required-motor-power-and-rotor-size-with-respect-to-the-wei
21. https://quadcopterproject.wordpress.com/static-thrust-calculation/
22. http://physics.stackexchange.com/questions/160435/relation-between-power-voltage-and-current-when-one-or-more-of-them-is-amplifie
23. C. Horne, “Expandable Open Source Autonomous Quadcopter,” *The UNSW Canberra at ADFA Journal of Undergraduate Engineering Research*, **6**, No. 1, (2013).