Persistent UAV Security Presence Service: Architecture and Prototype Implementation

Hyorin Park, Brad Hyeong-Yun Lee and James R. Morrison, Member, IEEE

1 Department of Industrial and Systems Engineering, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea.

Corresponding author (Tel : +82-42-350-3127; E-mail: james.morrison@kaist.edu)

Abstract: While unmanned aerial vehicles (UAVs) are a promising technology, they suffer from fundamental fuel and payload restrictions. A system of UAVs can be used to address these limitations. In this paper, we consider an automated system of UAVs seeking to provide an uninterrupted, or persistent, security presence to customers in an outdoor environment. We define the goals of such a system and provide architecture to achieve them. Various components such as the central director, mission planner and UAV service stations are introduced and analyzed. Following the architecture, we constructed prototypes of each component and implemented a small scale outdoor experiment. The experiment included receiving customers’ orders, algorithmic implementation of UAV tasks and UAV control.

Keywords: Autonomous system, Persistence, Robotic operating system (ROS), Service station, Unmanned aerial system (UAS)

1. INTRODUCTION

Technologies, infrastructures and regulations for unmanned aerial vehicles (UAVs) have advanced such that UAVs are being considered for various commercial applications in addition to military ones. One application of interest in both military and commercial domains is surveillance. In this domain, security escort and security patrol are important application candidates for UAVs.

Security escort and security patrol missions conducted with UAVs are expected to more flexibly cope with unexpected events and blind spots in comparison to fixed cameras. However, to faithfully conduct security missions, UAVs must be capable of continuous and long-term operation. UAVs appropriate for commercial or reasonable cost missions possess limited fuel and payload capabilities. As such, to provide quality security or patrol services, and to serve multiple customers simultaneously, a system of UAV is essential.

In order to practically operate such system of UAVs, not only should the mission objectives be achieved, but they must be accomplished as efficiently as possible. Efficiency can be achieved in part by near optimal assignment of system resources to the tasks at hand. With sufficient resources and by optimizing UAV task assignments, the missions can be conducted continuously, over the long-term, and well. Efficient task assignments can be derived and realized by optimization models such as the mixed integer linear program (MILP). In addition to task planning methods, various components such as middleware, central director, UAVs and automatic recharging stations enable persistent and efficient missions.

Recently, there have been many studies on various applications for UAVs and advancements in their component technologies. Two examples include the following. In [1], decision architecture, task scheduling models and algorithms for a system of UAVs to conduct fire detection was provided. The application of an indoor system of UAVs for use in a manufacturing facility was proposed in [2]. They introduced the system architecture and analyzed components including the scheduling method and control system.

Persistent surveillance mission via systems of UAVs have been considered in the literature. A decision making system to monitor the status of a system of UAV for persistent surveillance of fixed point, including indoor implementation, was detailed in [3]. In [4] and [5], a persistent surveillance system of UAVs for multiple fixed points in an indoor environment was studied. They provided system architecture and a scheduling algorithm to enable the operation of multiple UAVs. Experiments were conducted with UAVs and recharging stations.

There have been efforts to develop efficient task allocation methods for systems of UAVs. In [6], the authors formulated a MILP model for the task allocation problem based on a vehicle routing problem with time windows (VRPTW). They considered a single depot (launch site) with multiple UAVs and many target locations. The authors in [7] developed a MILP model for the optimal task allocation of multiple UAVs [7]. They constructed an exact MILP formulation for optimal solution and developed a suboptimal algorithm for real-time application.

Some research has focused on UAV surveillance missions with moving targets. In [8], a task allocation method for large systems of UAVs providing a security escort service was developed. The authors adopted a split job concept to enable continuous tracking. A MILP model and heuristic were used to assign tasks to multiple UAVs. In [9], an improved algorithm, components and small scale indoor implementation were presented.
There have been many supporting technologies developed to support UAV systems. The Ground Control Station (GCS) is important component to send commands, control and receive status from UAVs. In [10], authors designed architecture and implemented GCS software for a multi-UAV system using open source libraries. They also conducted a field test with two types of missions. Open source GCS software, such as QGroundControl [11], has been developed. QGroundControl provides a flight map displaying and mission planning for autonomous flight.

Fuel replenishment stations are an important component of persistent UAV systems. In [12], recharging stations were used to support a persistent hovering mission in a single location for three ROS controlled UAVs. To reduce the fuel replenishment time, battery replacement stations can be used. In [13], the authors focused on the design of a replacement station. They also analyzed the number of resources required and cost comparison in the system using a basic Petri net model. [14] detailed the design and construction of a battery replacement device and landing guide devices. The system was tested in an outdoor environment.

In this paper, we consider the development of a system of UAVs to provide a persistent security presence in an outdoor environment. The contributions follow.

- We develop architecture for system functions including customer order reception, information collection, UAV allocation and scheduling, UAV control and fuel service stations.
- We conduct analysis and logic development for each component based on proposed architecture.
- We develop prototypes of each component.
- Conduct a small scale implementation to demonstrate of key functionality.

The paper is organized as follows. In Section II, we propose a system for persistent UAV security presence (PUSP) to serve as a security escort. Section III introduces prototypes of key components. We discuss the small scale outdoor implementation in Section IV. Concluding remarks and future work are provided in Section V.

2. SYSTEM DESCRIPTION

We focus on providing a persistent UAV security presence for moving customers. The services to be provided include accompaniment of customers in unsafe areas (path following), patrolling designated routes and hovering (monitoring) specific areas. In order to provide such service, the mission must be handed off from one UAV to another at specific times and places. This allows weary UAVs to handoff the mission to a fresh UAV. To accomplish such handoffs efficiently, and when a large number of customers request surveillance missions, the task allocation problem becomes complex. A system with efficient planning algorithms is required.

In order to solve the task allocation problem for a system which includes path following missions, we adopt the split job concept of [6]. The split job concept divides a trajectory into segments and requires a single UAV to accompany a customer during that segment of their trajectory. Handoffs of the mission from one UAV to another are allowed only at the connection point between two trajectory segments. The split job concept discretizes the possible handoff locations. As such, a split job is the minimum task unit conducted by an UAV. See Figure 1. There, a single UAV can serve multiple customers in one flight but the customer can be accompanied by some UAV at all times. This allows UAVs to be used flexibly for several missions and enables them to travel to a fuel service station for replenishment.

2.1 Summary of proposed system

The physical components of a PUSP system are depicted in Figure 2. They include customers, a customer communication device (e.g., mobile phone and webpage), web server, main unit, UAVs and UAV controllers.

The system operation concept is as follows. Initially, customers request a UAV security presence through a
reservation webpage accessed via their mobile phone. The web server receives the request and sends it to the main unit which deliberates on the availability of system resources to accept the customer request. The main unit relies on data collected from the UAVs (via their controllers), previously stored geographic information and the customer’s requested route. A rolling horizon method from [15] is used in conjunction with a deterministic MILP task allocation model to determine new UAV tasks in response to the new request and assess if the new customer can be accepted. If the request is confirmed, the main unit sends a confirmation of the reservation to the customer. The tasks are distributed to the UAVs from the main unit through their controllers. The task allocation system anticipates when the UAV fuel levels will be nearly depleted (or the rolling horizon approach will adjust for unanticipated low fuel levels) and directs the UAVs to visit a fuel service station. Fresh UAVs are used to serve customers. Refer again to Fig. 1.

2.2 System architecture

The architecture of the system to accomplish the PUSP missions is shown in Figure 3. There are four components: a user interface, the main unit, UAV controllers and automatic replenishment service stations. Some details of each component are provided next.

1) User interaction

The user selects a desired departure point, destination and departure time through a reservation web page via an application on their mobile phone. This information is sent to the main unit through web server. As a result, it receives recommended route to the destination and the reservation availability such as confirmation, modified departure time, and rejection. Refer to Figure 4. The web server collects information from multiple users and transfers their requests to a data collector component in the main unit.

2) Main unit

The main unit consists of three parts. The core is a
mission planner which deliberates on customer requests and determines UAV task assignment. Preprocessing and post processing parts support the core in task such as receiving incoming tasks and UAV state information and delivering responses to customers and distributing UAV tasks.

Preprocessor (data collection and related processing)

The preprocessor collects customer information from the web server and receives information on the status of all UAVs. The customer information includes geographical information on their desired route (which is selected from among predetermined options) and time. The preprocessor converts the path into split jobs using an internal table prepared for that purpose. The information is converted to a convenient format for the mission planner.

Mission planner

As the core of the main unit, the mission planner conducts the essential service of determining the UAV task allocation. It solves the task assignment problem using efficient algorithm with the information gathered above. There are two methods internal to the mission planner that can be selected to solve the complex assignment problem: a MILP model and efficient heuristic. The MILP model can give an optimal solution, but it can be applied only to small models due to slow computational time. The heuristic can provide a non-optimal solution but can determine allocations with little computational time. It can be used for larger systems including many UAVs and tasks. Task assignment, time schedules and replenishing platform visit plans for each UAV are determined here.

Postprocessor (Action generate and release order)

The tasks assignments and schedules derived by the mission planner must be refined and converted to a format useful for each UAV. For example, the task allocation output from the central planner may dictate that task 1 is assigned to UAV 3 which is currently located at station A. However, for the UAV controller requires the following information: time for UAV3 to take off from station A, GPS location for the start of job 1, time to begin job 1 and GPS location for the end of job 1.

In the postprocessor, the action list which contains actions (represented as action ID, GPS coordinates and execution time) to be taken by UAVs, is made by action generating logic and stored. When the execution time is reached, the command is released to the ground control station.

3) UAV controlling system

To monitor the status of each UAV and conduct the actions dictated by the main unit, each UAV controlling unit consists of a relaying part and Ground Control Station (GCS) software. When the command is issued by the main unit, the relaying part stores the command and executes it at the right time. The control software manages the UAVs and directs them to conduct the corresponding action. At the same time, it receives status such as current location (GPS coordinates), battery level, and video from the UAV. In addition, it reports the same information to the main unit. The time scale diagram for each component in the system architecture is provided in Figure 5.

4) Automatic replenishing service station

To achieve persistent operations, fuel replenishing is an essential required function. It is done by the autonomous
replenishing service station. If a UAV is present on the station platform, the station is not available for other UAVs, so the availability must be reported to the system. This is can be done by direct communication between station and the main unit. However, in order to reduce the complexity of the system, the status of the occupying UAV may be used to infer the status of the station.

3. PROTOTYPES OF KEY COMPONENTS AND SMALL SCALE IMPLEMENTATION

We developed prototypes of key components in order to conduct a small scale implementation of a PUSP system in an outdoor environment. In this system, we assumed that patrol missions based on GPS coordinates alone are sufficient. For safety, ease of use and low cost we implement our system using AR Drone 2.0 UAVs.

3.1 Tasks reception (user interaction)

Considering the purpose of this service system, a customer must be able to select their desired route. To implement our small scale experiment, it is assumed that the customer’s starting points and destinations are limited to 4 specific locations, and the routes and its split jobs to each location are predetermined. In addition, it is assumed that all requests are not rejected and all UAV will proceed along their routes without disturbance.

Under the above conditions, we develop a reservation web page and mobile application as user interfaces for receiving requests from customers. See Figure 4. This allows the customer to select starting point and destination among 4 possible locations and set the departure time in 1 minute intervals. After reservation, the customer receives the recommended route and confirmation. All requests are stored in a web server which transfers the customer information such as customer ID (task identifier), starting point, destination, and departure time to main unit.

3.2 Main unit

We constructed a central unit as proposed here that includes three main parts. The preprocessor collects customers’ request (point of start, destination and departure time), information of UAVs (current location, and battery level), and split jobs information.

The mission planer determines the task allocation using the Sequential Tasks Allocation Heuristic (STAH) algorithm from [6]. This algorithm is considered characteristic of the split jobs concept and derives near optimal solutions in very fast computational time. This property makes it suitable for the event based rolling horizon concept which update when an events occur (e.g., new customer requests). Consequently, the mission planer is able to address uncertainties such as unexpected customer requests. Although the mission planer can deliberate and generate a new schedule frequently, we set it to update every 6 seconds. This prevents the inefficiency of UAV operation due to frequent mission changes.

At the postprocessor, lists of actions for each UAV are generated. The action generator disassembles the job assignment list and redefines them as detailed actions for the UAVs. The UAV actions for the security escort missions using GPS trajectories are defined as follows.

- A0 : Idle (on the platform)
- A1 : Take-off
- A2 : Go to the next designated GPS coordinate
- A3 : Stop at the designated GPS coordinate and hover
- A4 : Seek the station target
- A5 : Flight along designated path
- A6 : Landing
- A7 : Emergency landing or error
- A8 : Continue current action

Figure 6 provides the logic that generates the action list from an assignment plan. The action list is continuously updated every 6 seconds along with the mission planer. It is stored in the main unit and commands are released to the ground control station at the designated times.

3.3 UAV controller

Each laptop serves as a controlling unit and is connected to single UAV via the AR Drone 2.0 onboard Wi-Fi network. Each laptop is connected to the central unit to receive commands and transfer UAV state information. We used ROS (Robot Operating System) package as our ground control station software to conduct missions as directed by the central unit. We created an automated system to input waypoints to the control software. The UAVs follows the orders when the mission time arrives.

We developed an ROS package using the existing ROS tracking packages, ardrone_autonomy [16] and tum_ardrone [17]. The first ROS package supports GPS waypoint tracking functionality and the second package...
provides a convenient GUI. We extended tum_ardrone (which relies on ardrone_autonomy) to develop a new ROS package which is compatible with our multi-UAV system.

The developed ROS package as UAV controller receiving directives from the central unit as follows: first, it receives missions such as takeoff, track, and land from the central unit with designated mission times. Second, when the mission time arrives, the ROS package directs the UAV to conduct the mission. Reading missions provided by the central unit and conducting missions is the main function of the controller. The controller updates its mission status every second.

3.4 Automatic replenishing service station

Fuel supplementation is essential for PUSP service. We created a recharging service station, which is motivated by [7]. The battery recharge station hosts an onboard OEM charger for the AR Drone 2.0. The four pins of the OEM charger are extended and connected to a copper plate. The AR Drone 2.0 UAV was modified to receive power from the copper plates. Wires are connected from electrical contact on each foot of the UAV to the battery terminals. When a UAV lands on the copper plates, each pad provides a different voltage to the UAV battery terminals. The UAV must be in the correct orientation and position to charge. However, wind and environmental factors of the outdoor environment can make it a challenge for an AR Drone 2.0 UAV to land precisely. To solve the problem, the platform offers a spacious landing pad for easy landing

When the UAV lands inside the platform, an ultrasonic sensor detects the UAV presence and walls on the edges of the platform slide toward the interior and push the UAV to the center. In addition, rollers are attached to the walls to reduce friction and adjust the orientation of the UAV easily. A UAV attempting to land on the station will have already achieved a roughly correct orientation during the station seeking procedure. Consequently, the repositioning devices enable the UAV to be positioned within a centimeter of the exact desired position.

The charging procedure is as follows.

1. When the UAV lands on the station, the platform is activated by an ultrasonic sensor.
2. The edge walls move sequentially so that the UAV is not jammed during the pushing action.
3. When the UAV is in position, the charging device is activated.
4. The edge walls return to their original position to allow the UAV to depart.

3.5 Implementation

We conducted a small scale implementation. The purpose of this implementation was to ensure that the components described above function collectively. We also seek to verify that the algorithm developed in [13] can be used practically.
The description of the test is as follows. The test was conducted in an outdoor environment near our lab. It is assumed that there are four locations that customers can choose. The path and its split jobs when two of these four are selected are predetermined as shown in Figure 8. However, the schedule of the UAVs is not predetermined and stored in the GCS, but it is calculated by the main unit when the customer requests their order. Since the implementation is a small size test, it is assumed that the capability of UAVs is limited to check hand off issue. Consequently, we set the maximum flight time of a UAV to 600 seconds. Also, in the STAH algorithm, the unit time which indicates the time to perform one split job is set to 1 minute. We used AR. Drone 2.0 UAVs from Parrot with onboard GPS module. The ROS package described in Section 3.3 was used as the ground control station software. The main unit was hosted on its own laptop and one laptop was provided to host the GCS for each UAV. All computers connected to each other by a router. Further, UAVs are conneted via WIFI with different channel to avoid signal interference. The test environment is shown as Figure 9.

The experiment proceeded as follows. The customer’s request may come randomly, but for example, a customer requested a patrol mission from location 1 to 3 through the mobile device. Another customer requests an additional mission from location 4 to 2 at the same time. These orders are transmitted to the main unit via the web server. The STAH algorithm can create a schedule by solving problem in less than one second. It uses a time based rolling horizon method to check input data repeatedly every 6 seconds to avoid frequent changes to the plan. Figure 10 and Table 1 provide the schedule generated by the STAH algorithm. These schedules are decomposed into actions to be performed each time and then they are transmitted to each controller. The controller sends commanding orders to the developed ROS package which manages the UAV to conduct its task at the given time. A hand off occurred between location 1 to 3 and all of the patrol missions were conducted.

**CONCLUDING REMARKS**

The limited fuel capacity of UAVs hinders the prosecution of continuous and long-term missions. This can be overcome by a cooperative system of UAVs. Here we focused on persistent UAV security presence (PUSP) and constructed system architecture for such a service. We developed prototypes of each component and conducted a small scale demonstration in an outdoor environment.

In order to provide a realistic PUSP service to commercial or military application, much future work remains. A larger scale experiment implementation should be conducted. In order to make it a practical system, customer following via vision tracking technology should be incorporated.

**REFERENCES**


