Actual Flight Movements of Electric Helicopters for Making a Disaster Area Monitoring System

Takuya Saito*, Kenichi Mase**
* Research Institute for Natural Hazards & Disaster Recovery, Niigata University, Japan
** Academic Assembly, Niigata University, Japan

Article Info

Article history:

Received Sep 9, 2013
Revised Feb 2, 2014
Accepted Feb 26, 2014

Keyword:

AR.Drone
Disaster
Electric helicopter
Electric vehicle
Monitoring system

ABSTRACT

This paper describes the actual flight movements of electric helicopters and how to implement a disaster area monitoring system using electric ground vehicles and electric helicopters. A quad-rotor helicopter and a hex-rotor helicopter were used in the proposed system, specifically, the AR.Drone 2.0 and the DJI Innovations frame kit with an auto pilot system, respectively. We developed a software framework for the AR.Drone 2.0, which makes it possible to obtain flight data and images and to control automatic flight operations by computer. The relationship between the flight parameters and the real flight movements of the AR.Drone 2.0 were experimentally investigated. In conclusion, we found that the AR.Drone 2.0 with a software framework and a developed hex-rotor helicopter are sufficiently effective in implementing a disaster area monitoring system using electric ground vehicles and electric helicopters.

Corresponding Author:

Takuya Saito
Research Institute for Natural Hazards and Disaster Recovery, Niigata University,
8050 Ikarashi 2-no cho, Nishi-ku, Niigata-shi, Niigata, 950-2181, Japan
Email: takuya@toki.waseda.jp

1. INTRODUCTION

After a large-scale disaster, such as a major earthquake, it is very important to assess the situation of the disaster site as soon as possible. To achieve this, various disaster area monitoring systems have been proposed. For example, some use the sensor network technique; however, this type of monitoring system can only monitor disaster areas where cameras were installed in advance. Therefore, we developed and proposed a disaster area monitoring system using an electric ground vehicle (EV) that can drive around the disaster site to survey the situation. This EV has a wireless network system and many sensors, including a video camera and a GPS system..

In order to communicate with each other, the EVs are connected via wireless network in ad-hoc mode, such as the electric vehicular ad-hoc network (EVANET) [1][2]. However, this type of monitoring system can only survey areas that are visible to the driver, and not places hidden from the driver, such as the other side of a high wall or the roof of an apartment building.

Some studies used unmanned aerial vehicles (UAVs) equipped with cameras to survey large disaster areas from the sky. For example, Ohminato *et al.* observed the summit areas of active volcanoes using a UAV [3]. Suzuki *et al.* proposed real-time hazard map generation using a small UAV [4]. Alexis *et al.* proposed aerial forest fire surveillance using UAVs [5]. However, certain types of UAVs cannot be operated for a long time, because battery capacity is limited. An electric helicopter (EHs) can fly for approximately ten minutes only. For this reason, almost no EH can travel long distances.

Therefore, we propose a disaster area monitoring system using electric ground vehicles and electric helicopters [6][7]. Because EVs can run long distances, we can go to a disaster site using an EV and then launch the EH onsite. The EV serves as the aircraft carrier for the EH. It charges the battery of the EH and transmits the monitoring data obtained by the EH to other EVs or to base stations by using a wireless network, such as EVANET. The EH can either fly above the EV by manual operation or track a moving EV automatically by computer control.

A monitoring system that combines electric ground vehicles and electric helicopters has several advantages over monitoring systems that use only EVs or only EHs. Our disaster area monitoring system using electric ground vehicles and electric helicopters is at the early stage of development. In this paper, we describe the development of the software framework for the AR.Drone 2.0, the flight-data obtained by our prototype, and the flight performance of the hex-rotor helicopter that we developed.

This paper is organized as follows: In Section II, we describe the specifications of electric helicopters required by the disaster area monitoring system. In Section III, we discuss the software framework for the AR.Drone 2.0 and how it can be used to develop a monitoring system [8]. In Section IV, we describe how to control the flight of the AR.Drone 2.0. In Section V, we show the experimental results, including the flight data of the AR.Drone 2.0 and the flight performance of the custom hex-rotor helicopter. Section VI concludes the paper.

2. REQUIREMENTS AND SPECIFICATIONS OF ELECTRIC HELICOPTERS

We adopted electric helicopters in our disaster area monitoring system and decided to use an EV to provide the energy needed by the helicopters. For this study, the electric helicopter must satisfy various requirements as follows:

- The helicopter must be easy to operate without special training.
- The helicopter must be controllable by computer via a wireless network.
- The helicopter must be able to send real-time camera images via a wireless network.
- Using GPS or image analysis, the helicopter must be able to fly above an electric ground vehicle and track it automatically.

A helicopter that satisfies all of these demands is not currently available commercially. However, we found the AR.Drone 2.0 helicopter, which satisfies most of these demands. Therefore, we adopted the AR.Drone 2.0 as one of the EHs for the disaster area monitoring system. An AR.Drone 2.0 is shown on the left side of Figure 1.

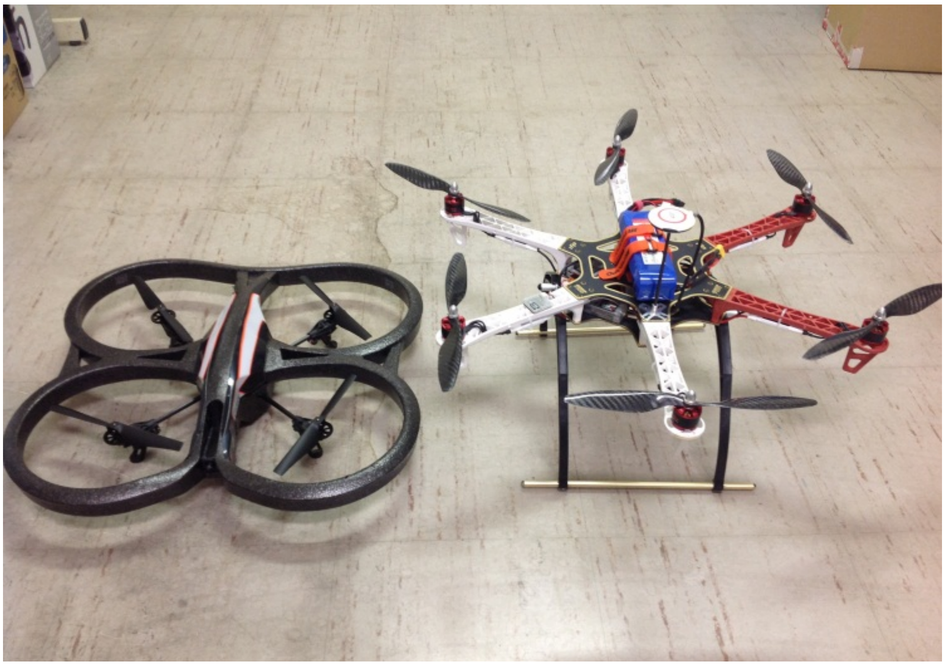

Figure 1. AR.Drone 2.0 and the hex-rotor helicopter

The AR.Drone 2.0 has many excellent features, including a Linux-based embedded system. It has an IEEE 802.11 b/g wireless network card, which can connect to and communicate with personal computers via an ad-hoc network. It has several useful sensors, including a high-definition video camera, three-axis gyroscopes, three-axis accelerometers, three-axis magnetometers, a pressure sensor, and an ultrasound sensor for ground altitude measurement. The AR.Drone 2.0 has a publicly available set of APIs that can be used to programmatically operate the helicopter via wireless network; therefore, we developed computer software to do so.

Because the AR.Drone 2.0 has the defects mentioned above, we decided to develop another helicopter – a hex-rotor helicopter using the DJI Innovation F550 frame kit and the WooKong-M auto pilot system [9]. The hex-rotor helicopter is shown on the right side of Figure 1. This hex-rotor helicopter satisfies all of our requirements, and it can be operated very easily using its auto pilot system.

3. DEVELOPMENT OF THE MONITORING SYSTEM USING AR.DRONE 2.0

Parrot, Inc. developed an SDK for the AR.Drone, and the source code is open to the public. However, this SDK consists of a large number of files, and its structure is too complex to understand with insufficient documentation. Therefore, the source code is too difficult to modify to implement the functions that we need. In addition, this SDK was primarily designed to test functions during the development of the AR.Drone; therefore, it is not suitable for our monitoring system. Instead, we decided to develop framework software to control the AR.Drone 2.0.

The AR.Drone 2.0 can connect to personal computers via its IEEE 802.11 b/g wireless network in ad-hoc mode. The IP address of the AR.Drone 2.0 is 192.168.1.1. The IP address of the personal computer that connects to the AR.Drone 2.0 is assigned automatically by the DHCP server from 192.168.1.2 to 192.168.1.5. The AR.Drone 2.0 uses several ports to communicate with the computers. The communication ports are shown in Table 1.

The AR.Drone 2.0 uses UDP port 5556 to receive flight operation commands. Every 30 ms, the personal computer sends AT commands as ASCII strings to the AR.Drone 2.0 to indicate flight movements, such as takeoff, landing, ascend, descend, pitch, roll, and yaw, and to configure the settings of the AR.Drone 2.0. Navdata, such as velocity, altitude, pitch, roll, and yaw, are obtained by the AR.Drone 2.0 through its sensors and sent to the personal computer through UDP port 5554 about 15 times per second. The AR.Drone 2.0 also sends video data through TCP port 5555. This video data is encoded in the H.264 or MPEG4 format with Parrot’s original header called PaVE. Finally, TCP port 5559 is used to send any configuration data for the AR.Drone 2.0.

This software framework for the flight control of the AR.Drone 2.0 was developed on MacOS X 10.8.2 Mountain Lion. The structure of this software framework is shown in Figure 2. The software framework is composed of five threads, because the data at each port need continuous processing.

Table 1. Communication Ports of AR.Drone 2.0

Port	Direction	Protocol	Description
5554	Drone to PC	UDP	Flight data (navdata)
5555	Drone to PC	TCP	Video data
5556	PC to Drone	UDP	Operate command (AT)
5559	Drone to PC	TCP	Control port

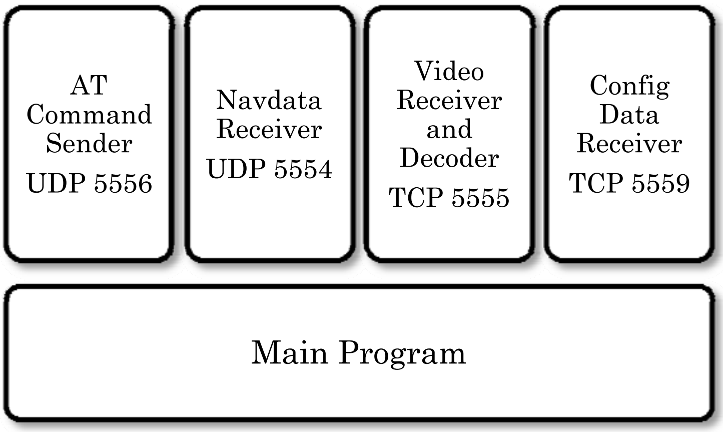

Figure 2. Structure of the software framework

4. HOW TO CONTROL THE FLIGHT OF AR.DRONE 2.0

4.1. Flight Data of AR.Drone 2.0

The software framework that we developed enables us to receive navdata, including the flight data, from the AR.Drone 2.0. An example of actual flight data received from the AR.Drone 2.0 is shown in Figure 3.

We set the AR.Drone 2.0 to navdata_demo mode; therefore, it sends navdata about 15 times per second. Navdata includes battery level, pitch, yaw, altitude, three-axis velocity, and other information. Figure 4 shows the assignment of the x, y, z, pitch, roll, and yaw axes of the AR.Drone 2.0.

4.2. Flight Parameters of AR.Drone 2.0

The flight of the AR.Drone 2.0 is controlled by AT commands, which are ASCII strings. The AT*REF command is used to control the basic behavior, such as takeoff, landing, emergency stop, and reset. The AT*PCMD command controls the flight motions, such as pitch, roll, and yaw. The value of each parameter is a percentage of the maximum bending angle, which is set to another configuration parameter called the Euler angle max (EAM). Therefore, a target-bending angle can be calculated by the following:

$$\textit{Target Angle}\ (\text{rad}) = \textit{EAM}\ (\text{rad}) \times \textit{Pitch}\ (\%) \tag{1}$$

For example, if we set the value of the EAM parameter to 0.5 and the value of the pitch parameter to 0.3, the target-bending pitch angle is 0.5 rad × 30% = 0.15 rad.

The developer manual of the AR.Drone 2.0 [10] explains these parameters as above. However, when we set these two parameters to a specific target value, the drone did not fly at that target angle. Therefore, we performed a flight experiment with the AR.Drone 2.0 to investigate the actual effect of these parameters.

```
Navdata Received! packet No=360 size=500 header=1432778632
state=260048080 seqNum=368370 visionFlag=0
id=0 size=148 -Navdata_demo- ctrl_state=131072 vBat%=83 phi=-
133 psi=-22267 altitude=0 vx=0 vy=0 vz=0
id=16 size=328 -Vision Detect-
id=65535 size=8 -Check Sum- checksum=7432
```

Figure 3. Actual flight data received from the AR.Drone 2.0

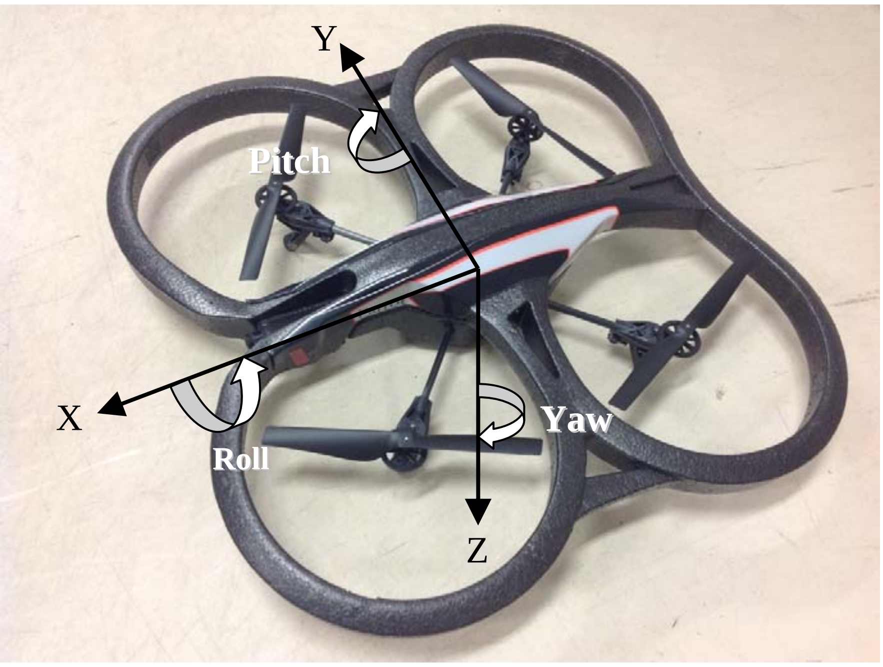

Figure 4. X, Y, Z axis and pitch, roll, yaw axis direction of the AR.Drone 2.0

5. EXPERIMENT

5.1. Experimental Environment

Our flight experiment investigated the relationship between the parameter values and real flight movements of the AR.Drone 2.0. The experimental environment is shown in table 2.

Table 2. Experimental Environment

OS	MacOS X 10.8 Mountain Lion
Computer	Apple MacBook Air 13-inch Mid 2012
Processor	Intel Core i7
Memory	8GB DDR3L SDRAM
Storage	SSD 256GB
RC Helicopter	AR.Drone 2.0
AR.Drone Hardware	Version 2.1
AR.Drone Software	Version 2.2.9

5.2. Flight Experiment of AR.Drone 2.0

We chose the target-bending pitch angle to be 0.2 rad. The target-bending pitch angle can be calculated by formula (1). We changed the combination of the EAM value and the pitch value four times, but all of the combination values resulted in the same target-bending angle, that is, 0.2 rad. The combinations of the values of the EAM and the pitch parameters are shown in Table 3.

The flight data of the AR.Drone 2.0 can be obtained by referring to Navdata, which was sent every 65 ms. The experiment was performed five times, and the final result was obtained by averaging them. In the experiment, we launched the AR.Drone 2.0 and, few seconds later, we set the EAM and the pitch values.

Figure 5 shows the relationship between the parameter values and the actual flight pitch angles sent from the AR.Drone 2.0. We set the EAM and the pitch values at time = 0. All combinations of EAM and pitch angles were expected to result in the same target angle, which is 0.2 rad. However, the experiment showed that the AR.Drone 2.0 flies at different angles, which are in proportion to the EAM values. When EAM is 0.2, the flight pitch angle is around 0.2 rad. Likewise, when EAM is 0.3, 0.4, and 0.5, the flight pitch angle is around 0.3, 0.4, and 0.5 rad, respectively.

Table 3. Combinations of Euler Angle Max and Pitch Parameters

Euler Angle Max (rad)	Pitch (%)	Target Angle (rad)
0.2	1.0	0.2
0.3	0.65	0.195
0.4	0.5	0.2
0.5	0.4	0.2

Figure 6 shows the relationship between the parameter values and the actual flight speeds. When EAM is 0.20, the actual flight speed increases gradually. However, when EAM is 0.30, the actual flight speed increases somewhat rapidly and overshoots slightly. When EAM is 0.40 and 0.50, the flight speed increases very rapidly and overshoots greatly.

The results are as follows:

- The flight angle does not match the Target Angle calculated by (1), except that the EAM value is 0.2.
- The flight angles agree somewhat with the EAM values.
- When the EAM value is 0.2 and 0.3, the actual flight pitch angle is stable.
- When the EAM value is 0.4 and 0.5, the actual flight pitch angle is unstable.
- When the EAM value is 0.2, the flight speed increases gradually.
- When the EAM value is 0.3 or higher, flight speed overshoot is observed.

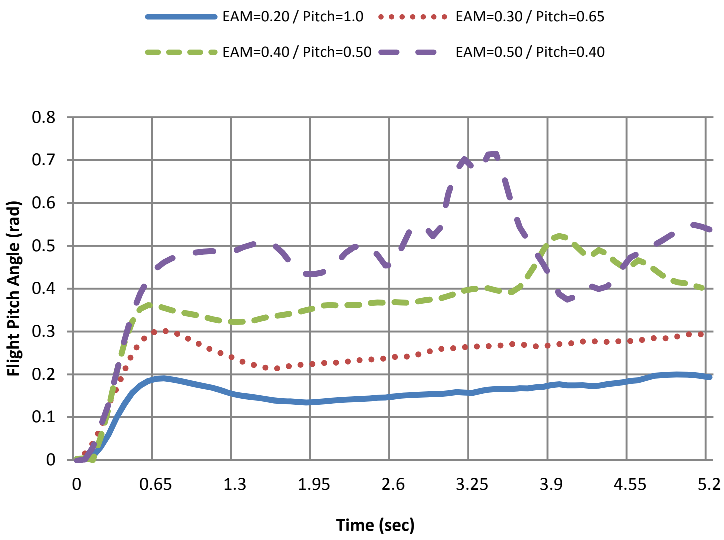

Figure 5. Relationship between parameter values and flight pitch angles

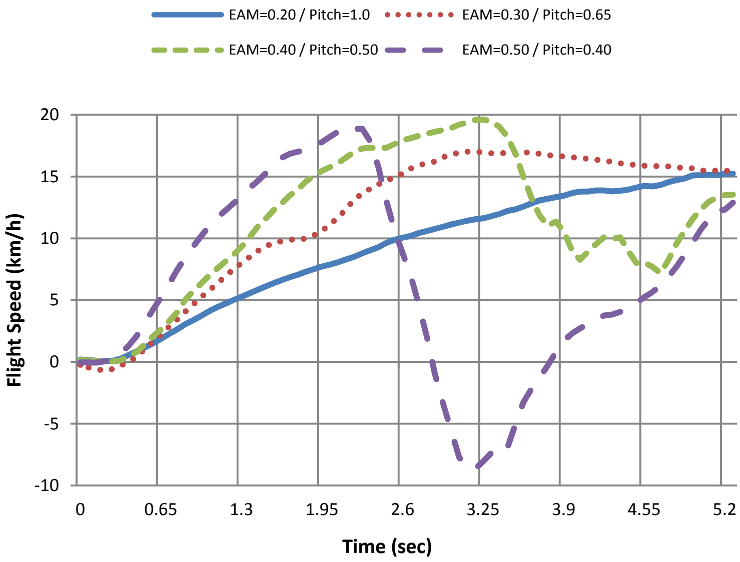

Figure 6. Relationship between parameter values and flight speeds

In conclusion, we found that the actual flight pitch angle agrees somewhat with the EAM value, and the ideal EAM value is 0.2, because, when EAM is 0.3 or higher, the flight angle and the flight speed become unstable. We also found that the pitch value has no effect on the flight angles when the value is 0.4 or higher.

5.3. Flight Experiment with EAM Value Fixed at 0.2

We decided to set the EAM value at 0.2, because we found that 0.2 is the ideal EAM value based on the above-mentioned experiment. However, the results also show that the pitch value has no effect on the actual flight angle. We suspected that a pitch value of 1.0 does not mean 100%, and that a value of 0.4 or higher means over 100%. Therefore, to determine what the pitch value means, we performed a flight experiment where the EAM value was fixed at 0.2 and the pitch value changed to 0.1, 0.2, 0.3, 0.4, and 1.0.

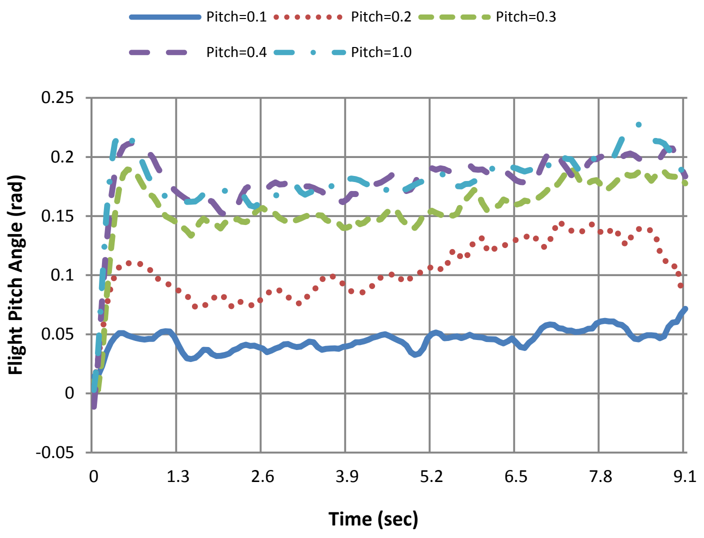

Figure 7. Relationship between pitch value and flight pitch angle

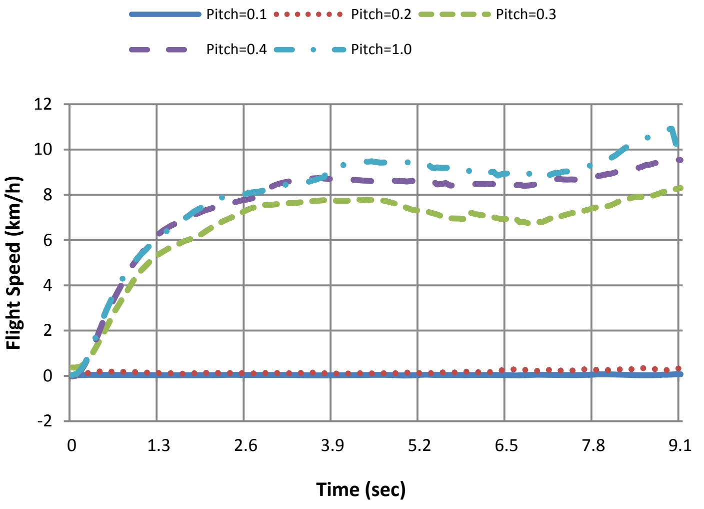

Figure 8. Relationship between pitch value and flight speed

Figure 7 shows the relationship between pitch values and flight pitch angles. When we changed the pitch value to 0.1, 0.2, and 0.3, the flight pitch angles changed linearly. However, when we set the pitch value to 0.4 or higher, the actual flight pitch angle did not change linearly; instead, it seemed to exceed the maximum level. This result shows that the value of the pitch parameter is valid when it is less than or equal to 0.3. We also observed that when the pitch value is 0.1, the flight pitch angle is around 0.05 rad, which is approximately 25% of the EAM value of 0.2 rad. Likewise, when the pitch values are 0.2 and 0.3, the flight pitch angles are approximately 0.10 rad and 0.15 rad, respectively, which are approximately 50% and 75%, respectively, of the EAM value.

Figure 8 shows the relationship between pitch values and flight speeds. The flight speed can be measured when the pitch value is greater than or equal to 0.3. Although the measured flight speeds are around 0 when the pitch values are 0.1 and 0.2, the AR.Drone 2.0 actually flies forward. This means that the flight speed data in the navdata is unreliable, especially when the pitch value is less than or equal to 0.2.

In conclusion, the experimental result shows the following:

- The EAM value is the maximum bending pitch angle.
- The pitch value is the percentage of the maximum bending pitch angle.

- A pitch value of 0.1 corresponds to about 25% of the EAM value.
- Valid pitch values are from 0 to 0.3.

These experimental results revealed that the actual target-bending flight pitch angle of the AR.Drone can be calculated by the following:

$$\textit{Target Angle}\ (\text{rad}) = \textit{EAM}\ (\text{rad}) \times 25 \times \textit{Pitch}\ (\%) \tag{2}$$

5.4. Flight Performance Measurement of the Hex-rotor Helicopter

We measured the flight performance of the hex-rotor helicopter that we developed, including maximum speed, highest altitude, maximum ascending speed, payload, and battery runtime. The measured performance results are shown in Table 4. All these measured flight performances satisfy our requirements of the helicopter we need in order to implement a disaster area monitoring system.

Table 4. Performance Measurement Results of the Hex-rotor Helicopter

Measurement Item	Measured Value
Maximum speed	53 km/h
Highest altitude	800 m
Maximum ascending speed	5.3 m/s
Payload	1000 g
Battery run time	16 minutes

6. CONCLUSION

In this paper, we proposed a disaster area monitoring system using electric ground vehicles and electric helicopters. We developed a software framework for the AR.Drone 2.0 to implement this system. We obtained the flight data of the AR.Drone 2.0 and investigated the relationship between flight parameters and flight data. We found that the EAM value is the maximum bending angle, that the pitch value is a percentage of the maximum bending angle, and that a pitch value of 0.1 is approximately 25% of the EAM. The result of our experiment proved that the ideal value for the EAM is 0.2, and that valid pitch values are 0 to 0.3. We found that the flight speed data in the navdata cannot be trusted. We found that the actual target-bending flight pitch angle of the AR.Drone can be calculated by a formula using the EAM and pitch percentage.

We also developed a hex-rotor helicopter using the DJI Innovations frame kit and an auto pilot system named WooKong-M to satisfy our requirements for our disaster area monitoring system. At present, the hex-rotor helicopter is operated using a proportional radio control system. Our next step is to develop the embedded system and software that will enable flight operation by computers and to investigate its actual flight movements similarly.

This study constitutes a first step toward realizing a monitoring system using electric ground vehicles and electric helicopters. Much work is left to do in order to implement this monitoring system. For example, we have to study how to attain autonomous flight by computer control, how to program a flight to track an EV by GPS, and what kind of image analysis is useful to find victims of natural disasters. The challenge now is to identify these possibilities.

ACKNOWLEDGEMENTS

This work was supported by a JSPS Grant-in-Aid for Scientific Research, Grant Number 24246068. We would like to thank Mr. H. Machinaka, Mr. K. Okimura, and Mr. K. Imai for assistance with the measurement of flight data.

REFERENCES

[1] K. Mase, “Information and Communication Technology and Electric Vehicles – Paving the Way Towards a Smart Community,” *IEICE Transactions on Communications*, vol. E95-B, no. 6, pp. 1902-1910, 2012.

[2] T. Ohminato, *et al.*, “Volcano Observations using an Unmanned Autonomous Helicopter: Seismic and GPS Observations Near the Active Summit Area of Sakurajima and Kirishima Volcano, Japan,” *EGU General Assembly 2012*, pp. 8575, Apr. 2012.

[3] Y. Matsuda and K. Mase, “A Proposal of Ad Hoc Networks Using Mini-EVs,” *Technical Report of IEICE*, AN2011-29, pp. 69-74, Oct. 2011.

[4] T. Suzuki, *et al.*, "Real-time Hazard Map Generation using Small Unmanned Aerial Vehicle," *Proc. Of SICE Annual Conference*, pp. 443-446, Aug. 2008.

[5] K. Alexis, *et.al.*, "Coordination of Helicopter UAVs for Aerial Forest-Fire Surveillance," *Applications of Intelligent Control to Engineering Systems*, vol. 39, pp. 169-193, 2009.

[6] T. Saito, H. Machinaka, and K. Mase, "A Proposal of a Disaster Area Monitoring System," *Technical Report of IEICE*, vol. 112(239), pp. 64-68, Oct. 2012.

[7] K. Mase and T. Saito, "Electric-Vehicle-Based Ad Hoc Networking and Surveillance for Disaster Recovery," *The Ninth International Conference on Networking and Services (ICNS2013)*, pp. 87-93, Mar. 2013.

[8] Parrot, "AR.Drone 2.0," available from http://ardrone2.parrot.com/, accessed on Sep. 06, 2013.

[9] DJI Innovations, "DJI Innovations Flight Control Experts," available from http://www.dji-innovations.com/, accessed on Sep. 06, 2013.

[10] Parrot, "AR.Drone 2.0 developer guide SDK 2.0," ARDRONE open API platform, available from https://projects.ardrone.org/, accessed on Sep. 06, 2013.

BIOGRAPHIES OF AUTHORS

Takuya Saito received his B. E., M. E., and Dr. Eng. Degrees in Computer Science from Waseda University, Tokyo, Japan, in 2003, 2005, and 2011, respectively. In 2011, he was a research assistant in the Department of Computer Science and Engineering, Waseda University. Since 2012, he is an assistant professor in the Research Institute for Natural Hazards and Disaster Recovery, Niigata University. He has participated in many robot contests and he has won many awards in contests such as Open-R Techno Forum, Robo-One, and RoboCup. His research interests include humanoid robot control, movement and interactive communication, image analysis, augmented reality, 3D graphics, computer music, and unmanned aerial vehicles.

Kenichi Mase received the B. E., M. E., and Dr. Eng. Degrees in Electrical Engineering from Waseda University, Tokyo, Japan, in 1970, 1972, and 1983, respectively. He joined Musashino Electrical Communication Laboratories of NTT Public Corporation in 1972. He was Executive Manager, Communications Quality Laboratory, NTT Telecommunications Networks Laboratories from 1994 to 1996 and Communications Assessment Laboratory, NTT Multimedia Networks Laboratories from 1996 to 1998. He joined Niigata University in 1999 and is now Professor Emeritus, Niigata University, Niigata, Japan. He received IEICE Best Paper Award in 1994, the Telecommunications Advanced Foundation Award in 1998, and Best Paper Award, International Academy, Research, and Industry Association in 2013. His research interests include communications network design and traffic control, quality of service, mobile ad hoc networks, and wireless mesh networks. He was President of IEICE-CS in 2008 and Vice President of IEICE in 2011 and 2012. Prof. Mase is an IEEE and IEICE Fellow.