

## Autonomous Flight Control System of Quadcopter Based on Waypoint and its Application to Formation Control with Mobile Robot

Atsushi Fujimori<sup>\*1</sup>, Yu-uki Ukigai<sup>\*2</sup>, Shuheji Santoki<sup>\*3</sup>, Shinsuke Oh-hara<sup>\*4</sup>

<sup>\*1</sup>(Department of Mechanical Engineering, University of Yamanashi, Japan

<sup>\*2</sup>(Department of Mechanical Engineering, University of Yamanashi, Japan

<sup>\*3</sup>(Department of Mechanical Engineering, University of Yamanashi, Japan

<sup>\*4</sup>(Department of Mechanical Engineering, University of Yamanashi, Japan

Corresponding Author ; Atsushi Fujimori

### ABSTRACT

This paper presents an autonomous flight control system of a quadcopter ARdrone 2.0 only using the internal sensor. Since AR.Drone is designed as a quadcopter with four control commands, four velocity models are constructed by the step response experiments. The basic control laws for tracking and positioning control are combined with each other to construct the complex control system for required flight missions. Furthermore, the proposed autonomous flight control system is applied to a formation control with a mobile robot. The effectiveness of the proposed autonomous flight control system is demonstrated in experiments.

**Keywords** - Autonomous flight, formation control, UAV

Date Of Submission: 06-10-2019

Date Of Acceptance: 23-10-2019

### I. INTRODUCTION

Recently, a great deal of practical application of quadcopters to monitoring, inspection and measurement has been reported by [1] - [6]. There are two kinds of operations in quadcopters; manual operation and autonomous operation by PC or iPhone. It is desirable for use of quadcopters in several purposes to establish robust and reliable autonomous operation [1], [5]. The autonomous flight based on external sensors such as the GPS arrives at the practical level [1]. If autonomous flight without external sensors is realized, the use of quadcopter is more extended in several areas. A number of autonomous flight of quadcopters have been reported from various points of view. Feedback linearization [7], backstepping method [8], sliding mode control [9], nonlinear adaptive control [10] and PID compensation [11] were adopted to compensate nonlinearities of the quadcopters. Robust control [12], auto-tuning I-PD [13] and model predictive control [14] were suggested as the method that dealt with uncertainties of quadcopters. Furthermore, visual cameras were used for controlling the attitude of the quadcopter [15], [16].

This paper presents an autonomous flight control system of a quadcopter in which only internal sensors are used. The controlled quadcopter considered in this paper is AR.Drone 2.0 which have been developed by Parrot Inc. [17]. Comparing this paper with the researches which have been published, the advantages are given as follows. The motion of

the quadcopter is first modelled by four independent linear velocity models and basic control laws are designed in each model. Integrating these control laws, a complex flight control system is constructed for achieving the specified flight mission of the quadcopter. The time-delay due to the communication between the quadcopter and the PC is also included in the linear models.

The proposed flight control system is furthermore applied to a formation control with a mobile robot. The effectiveness is demonstrated by experimental results.



Figure 1: AR.Drone 2.0

## II. QUADCOPTER AND MODELING

### II.A. AR.Drone

Figure 1 shows AR.Drone 2.0 [6] which is used as a quadcopter in this paper. AR.Drone 2.0 has four propellers whose diameter is 200 (mm), and are driven by four independent electrical motors. The size of the body is 510 times 510 (mm), the height is 110 (mm). The weight including the battery is about 460 (g). The gyro, accelerometers and ultrasonic sensors are installed for estimating the flight velocity and the attitude angles. Furthermore, two pinpoint cameras are attached at the front and the bottom.

AR.Drone 2.0 is able to communicate with a PC or a mobile phone by means of the wireless LAN [7]. The sensor information of AR.Drone 2.0 is received from AR.Drone 2.0 to PC, while the control commands are sent from PC to AR.Drone 2.0. These communications are supported by the C++ programming environment.

### II.B. Velocity Model

The control command of a quadcopter corresponds to four rotor thrusts. Seeing from a point of view in the flight dynamics, it is possible to approximately treat the motion of the quadcopter as four independent motions; front-back direction ( $u$ -motion), horizontal direction ( $v$ -motion), vertical direction ( $w$ -motion) and rotational direction ( $r$ -motion) in a stable flight condition [19], [20]. Where  $u$ ,  $v$  and  $w$  are the forward, the lateral and the vertical velocities, respectively.  $r$  is the yaw angular velocity. The control commands of AR.Drone are given as the four commands for the velocities  $u$ ,  $v$ ,  $w$  and  $r$ . Then, four velocity models, named as  $u$ -,  $v$ -,  $w$ - and  $r$ -model, are constructed for the four control commands  $u_{com}$ ,  $v_{com}$ ,  $w_{com}$  and  $r_{com}$ . AR.Drone 2.0 contains internal controllers to stabilize the attitude of the body [17]. That is, the pitching motion with respect to the pitch angle  $\theta$  and its rate  $q$  in  $u$ -motion is controlled so as to be a linear stable second-order system. Similarly, the rolling motion with respect to the roll angle  $\varphi$  and its rate  $p$  in  $v$ -motion is controlled so as to be a linear stable second-order system. The velocity models to be considered in this study represent the input-output characteristics including the internal controllers.

In our previous study [22], the step response for each motion was examined to estimate the linear discrete-time models. As the result,  $u$ - and  $v$ -motion were estimated as the third-order models, respectively,  $w$ -motion was the first-order model and  $r$ -motion was the zeroth-order model. It was necessary to include the time-delay element due to the communication by the wireless LAN in each model. The time-delay was the eight sampling when the sampling time was  $T_s=0.05$  (sec).

**Table 1: Threshold values of saturation and dead zone**

	$u$ -model	$v$ -model	$w$ -model	$r$ -model
$x_{max}$	2.0	2.0	2.0	2.0
$x_{min}$	0.02	0.02	0.00	0.03

### II.C. Input Nonlinearity

There are two nonlinear properties in the control commands of AR.Drone 2.0; the saturation and the dead zone denoted as  $\text{sat}(\cdot)$  and  $\text{dz}(\cdot)$ , respectively [21], [22] which are caused by the constraint on the electric current to motors and the resolution on the control commands. They are given by

$$\text{sat}(x) = \begin{cases} -x_{max} & (x < -x_{max}) \\ x & (|x| \leq x_{max}) \\ x_{max} & (x > x_{max}) \end{cases} \quad (1)$$

$$\text{dz}(x) = \begin{cases} x & (|x| > x_{min}) \\ 0 & (|x| \leq x_{min}) \end{cases} \quad (2)$$

where  $0 < x_{min} < x_{max}$ . The threshold values in  $\text{sat}(\cdot)$  and  $\text{dz}(\cdot)$  were estimated as Table 1 [22].

## III. AUTONOMOUS FLIGHT CONTROL SYSTEM OF QUADCOPTER

### III.A. Basic Control Law

Considering practical operation of the quadcopter, the following flight strategies are needed; tracking control based on the specified flight velocity profile and positioning control to the specified positions. In the former, the flight with exact velocity is not necessarily required in this paper. Since the constructed models of the quadcopter are stable models, the tracking control laws are then given by static feedforward laws. Letting  $u_{ref}$  be the reference of the forward velocity in  $u$ -motion, the control command is given by

$$u_{com} = K_u u_{ref} \quad (3)$$

$$K_u = 1 / \lim_{z \rightarrow 1} G_u(z) \quad (4)$$

where  $K_u$  is the feedforward gain and  $G_u(z)$  is the pulse transfer function of the  $u$ -model. It is remarked that  $u_{ref}$  should be given by considering the input nonlinearities which mentioned in the previous section.

In the latter, on the other hand, the model for positioning control, called position model, is constructed by adding an integrator to the velocity model in the output side. The positioning control law is then given by the state feedback so as to stabilize the position model and to achieve the position to the referenced point. Since the sensor information from AR.Drone is transitional velocities;  $u$ ,  $v$  and  $w$  and the attitude angles;  $\varphi$ ,  $\theta$  and  $\psi$ , the positional information is obtained by numerically integrating the transitional velocities, while the angular velocity information is obtained by numerically

differentiating the attitude angle. The states for the time-delay are obtained by shifting the control command every sampling. Since an integrator is included in the position model, the positioning control is achieved without the steady-state positional error if the reference position is constant. Furthermore, it is necessary to take into account the input nonlinearities in the control law of the positioning control. Letting  $e_x$  be the distance to the specified position, the discrete-time control command at the  $k$ -th sampling is given by

$$\tilde{u}_{com}(k) = -F\mathbf{x}(k) - F_d\boldsymbol{\xi}(k) \quad (5)$$

$$u_{com}(k) = \text{dz}(\text{sat}(\tilde{u}_{com}(k))) \quad (6)$$

$$\xi_i(k) = u_{com}(k-i) \quad (i=1, \dots, 8) \quad (7)$$

where  $\mathbf{x}(k) = [-e_x(k) \quad u(k) \quad \theta(k) \quad q(k)]^T$  is the state vector,  $\boldsymbol{\xi}(k) = [\xi_1(k) \quad \dots \quad \xi_8(k)]^T$  is the auxiliary vector for the time-delay and  $\tilde{u}_{com}(k)$  is the control command by the state feedback.  $F$  and  $F_d$  are the state feedback gains which are designed in the manner of the LQR [23].

### III.B. Complex Control Based on Way Points

As a concrete method to perform an autonomous flight of quadcopter, multiple way points (WPs) are defined as the flight targets in the world coordinates. The objective of the flight is to travel the WPs in order, where the flight trajectory between WPs is assumed to be the straight-line. To do this, the four basic control laws are incorporated into the flight control system. The basic concept of the flight control system is given as follows.

(a) When the quadcopter is far from the targeted WP, the flight velocity is controlled to be constant. That is, the tracking control by the feedforward law given by Eqs. (3) and (4) is adopted.

(b) When the quadcopter is near to the targeted WP, the control law is switched to the positioning control by the state feedback law which are given by Eqs. (5), (6) and (7).

(c) In the straight-line flight, the control of  $r$ -motion is performed independently of other motion control.

### III.C. Switch of Control Law

This subsection shows the switch between the tracking and the positioning control laws. Letting ( $e_x, e_y, e_z$ ) be the elements of the distance to the targeted WP in the body-fixed coordinates, the control law in the forward ( $x$ ) direction is switched as follows.

```

if |e_x| > R_p1,
then mode(u) = "tracking", K_u = e_x / |e_x|
if |e_y| > R_p2,
then mode(v) = "tracking", K_v = e_y / |e_x|
else mode(v) = "positioning"
if |e_z| > R_p2,
    
```

```

then mode(w) = "tracking", K_w = e_z / |e_x|
else mode(w) = "positioning"
else mode(u) = "positioning"
    
```

Where  $R_{p1}$  and  $R_{p2}$  ( $R_{p1} > R_{p2} > 0$ ) are threshold values with respect to the distance. The former is the one for "far", while the latter is the one for "near".  $\text{mode}()$  means the control law of  $u$ -,  $v$ - and  $w$ -motion.  $K_u, K_v$  and  $K_w$  are the feedforward gains of the tracking control and are adjusted according to the distance ratio between the coupled motions. The similar control switch is also installed in the lateral ( $y$ ) and the vertical ( $z$ ) directions.

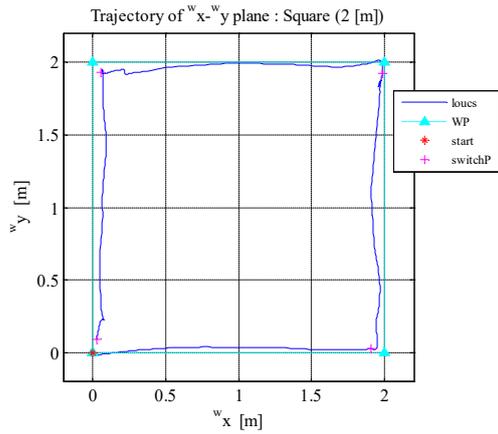
### III.D. Flight Experiment Based on WP

To evaluate the effectiveness of the autonomous flight control system which has been mentioned so far, flight experiments based on the multiple WPs were carried out. The positioning control in  $r$ -motion is applied to keep the yaw angle which the autonomous flight started.

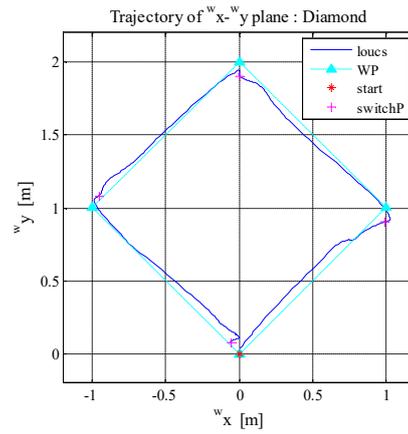
**1) Square:** Figure 2 shows the experimental result where four WPs were placed at the corners of a square whose length of the segment was 2 (m). The height was 0.8 (m) on the floor. The threshold values for the control switch were given by  $R_{p1} = 1.0$  and  $R_{p2} = 0.1$  (m). In Fig. 2(a), "switchP", denoted as magenta cross, means the point where the targeted WP is switched to the next. AR.Drone almost flew along with the lines between the WPs. The positional error at WPs was less than 0.05 (m) in the flight experiment. Figure 2(b) shows the time histories of the position ( $^w x, ^w y$ ) in the world coordinates, the velocities ( $u, v$ ) and the control commands ( $u_{com}, v_{com}$ ). When the distance to the target WP was longer than  $R_{p1}$ , the tracking control was selected because  $u_{com}$  or  $v_{com}$  were given constant (0.5) as shown in Fig. 2(b). When the distance to the target WP was shorter than  $R_{p1}$ , the control law was switched to the positioning control without any unstable responses. Figure 2(c) shows the time histories of the yaw angle  $\psi$ , the yaw rate  $r$  and its control command  $r_{com}$ . It is seen that AR.Drone was traveled between the four specified WPs with keeping the yaw angle error of 2 (deg).

**2) Diamond:** Figure 3 shows the experimental result where four WPs were placed at the corners of a diamond whose length of the segment was 1.414 (m). Although  $u$ -motion and  $v$ -motion were coupled with each other in this case, stable autonomous flight was realized.

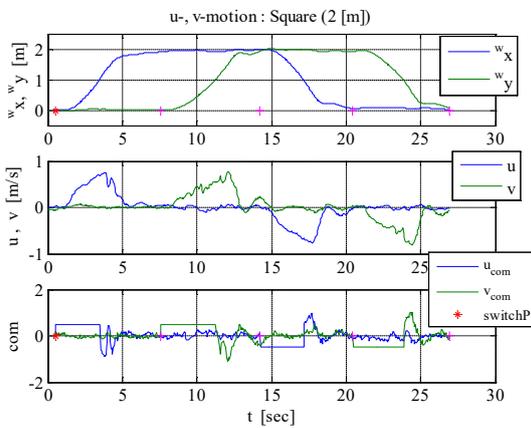
**3) Cubic:** Figure 4 shows the experimental result where four WPs were placed at the corners of a cubic whose length of the segment was 1.0 (m). The flight trajectory between WP with height varying did not become to be linear.



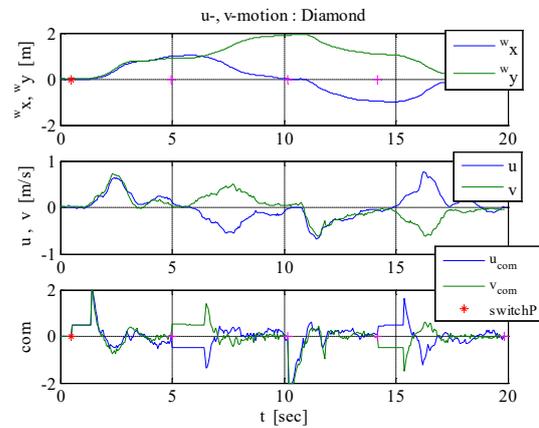
(a) Trajectory



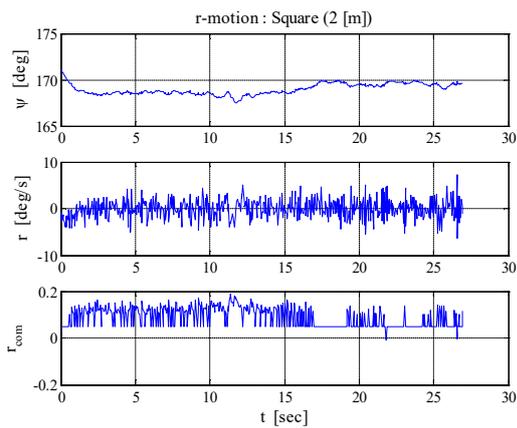
(a) Trajectory



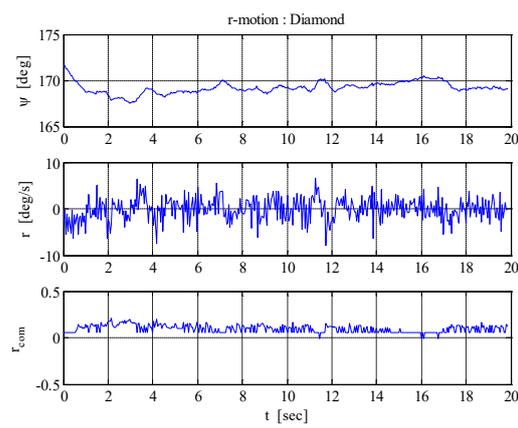
(b)  $u, v$ -motion



(b)  $u, v$ -motion



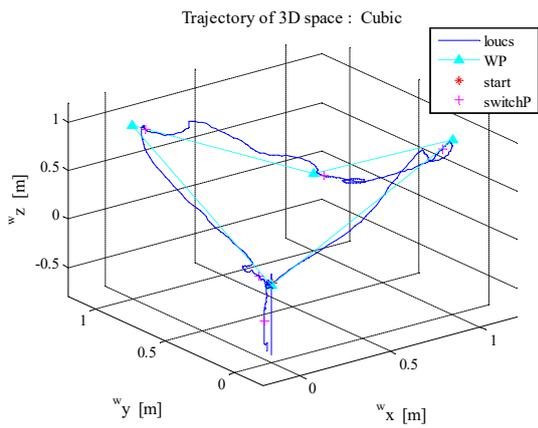
(c)  $r$ -motion



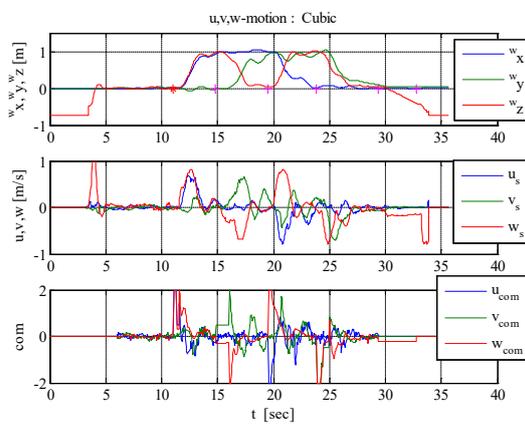
(c)  $r$ -motion

Figure 2: Autonomous flight 1: Square

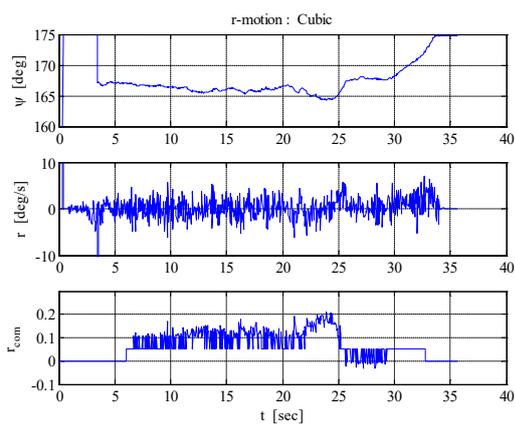
Figure 3: Autonomous flight 2: Diamond



(a) Trajectory



(b)  $u, v$ -motion

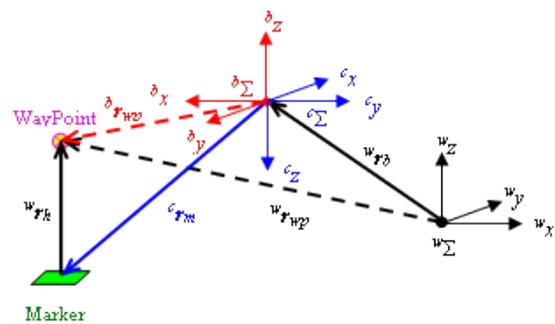


(c)  $r$ -motion

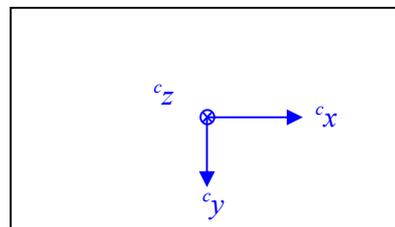
Figure 4: Autonomous flight 3: Cubic



Figure 5: Mobile robot Pioneer with aruco marker



(a) Aruco marker, AR.Drone and waypoint



(b) Camera coordinates  $c_\Sigma$

Figure 6: Coordinates with respect to AR.Drone



Figure 7: Formation control between AR.Drone and Pioneer

#### IV. FORMATION CONTROL WITH MOBILE ROBOT

To demonstrate the effectiveness of the proposed autonomous flight control system of quadcopter moreover, this section presents formation control between a quadcopter and a mobile robot. A scenario considered in this paper is given as follows. An aruco marker is attached on the top of a mobile robot, called Pioneer, as shown in Fig. 5. The quadcopter, AR.Drone detects the marker by the bottom pinpoint camera. When Pioneer moves in 2D-plane, AR.Drone follows Pioneer with keeping the specified distance in 3D-space. The aruco marker is one of augmented reality markers [24]. The programing environment is provided as a module in OpenCV [25]. To realize the formation control, the following functions have to be designed in the flight control system of AR.Drone.

**F-1:** AR.Drone detects the aruco marker by the bottom pinpoint camera.

**F-2:** AR.Drone flies to the point which is specified in advanced. In this paper, an WP is placed over the aruco marker. AR.Drone is controlled by the positioning control law and is shifted to hovering after arriving at the WP.

**F-3:** According to Pioneer's movement, AR.Drone autonomously flies with keeping the positional relation to Pioneer.

##### IV.A. Marker Detection and WP Set

AR.Drone is located it on the left side of Pioneer at the start. After takeoff, AR.Drone repeats zigzag flight while advancing to the right side to detect the marker (F-1). In the flight experiment, AR.Drone was able to detect the aruco marker with approximately 80 (%) of probability by the zigzag flight.

To realize F-2, an WP is set over the aruco marker. Figure 6 shows the relationship among the world coordinates  ${}^w\Sigma$  ( ${}^wx$ ,  ${}^wy$ ,  ${}^wz$ ), the body-fixed coordinates  ${}^b\Sigma$  ( ${}^bx$ ,  ${}^by$ ,  ${}^bz$ ) and the camera coordinates  ${}^c\Sigma$  ( ${}^cx$ ,  ${}^cy$ ,  ${}^cz$ ). Since the bottom pinpoint camera is attached at the center of gravity of AR.Drone, the origin of  ${}^c\Sigma$  is the same point as that of  ${}^b\Sigma$ . Then, the following relation holds.

$${}^cx = -{}^by \quad {}^cy = -{}^bx \quad {}^cz = -{}^bz \quad (8)$$

When the aruco marker is detected by the camera, the position vector of the marker in  ${}^c\Sigma$ , denoted as  ${}^c\mathbf{r}_m$ , and the attitude angle of AR.Drone are estimated. Letting  ${}^w\mathbf{r}_b$  and  ${}^w\mathbf{r}_h$  be the position vector of AR.Drone and the vector from the marker to the WP, respectively, the position vector of the WP in  ${}^w\Sigma$ , denoted as  ${}^w\mathbf{r}_{wp}$ , is given by

$${}^w\mathbf{r}_{wp} = {}^w\mathbf{r}_b + {}^w\mathbf{R}_b {}^b\mathbf{R}_c {}^c\mathbf{r}_m + {}^w\mathbf{r}_h \quad (9)$$

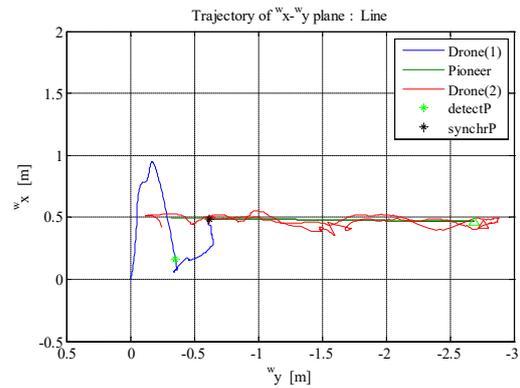
where  ${}^w\mathbf{R}_b$  and  ${}^b\mathbf{R}_c$  are the rotational matrices from  ${}^b\Sigma$  to  ${}^w\Sigma$  and from  ${}^c\Sigma$  to  ${}^b\Sigma$ , respectively. Furthermore, the position vector  ${}^b\mathbf{r}_{wp}$  in  ${}^b\Sigma$  given by

$${}^b\mathbf{r}_{wp} = {}^b\mathbf{R}_c {}^c\mathbf{r}_m + {}^w\mathbf{R}_b^{-1} {}^w\mathbf{r}_h \quad (10)$$

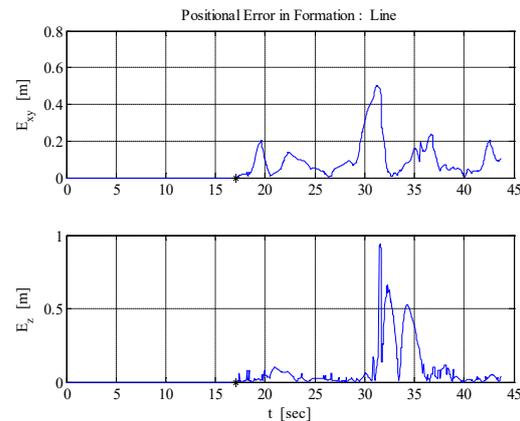
corresponds to the distance for the positioning control to the WP.

##### IV.B. Formation Control Experiment

The experiments of formation control between AR.Drone and Pioneer were carried out as shown in Fig. 7, where the trajectory of Pioneer was given by (i) line, (ii) square and (iii) circle and AR. Drone followed Pioneer by detecting the aruco marker. Figures 8 - 10 show the results of (i) - (iii), respectively. The height placing the WP over the aruco marker was given by 0.8 (m). Figure 8(a), 9(a) and 10(a) shows the 2D trajectory of AR.Drone and Pioneer in  ${}^w\Sigma$  where "Drone(1)", drawn by the blue line, means the trajectory of AR.Drone before detecting the aruco marker and "Drone(2)", drawn by the red line, means the one under the formation control. Applying the zigzag flight to AR.Drone, the

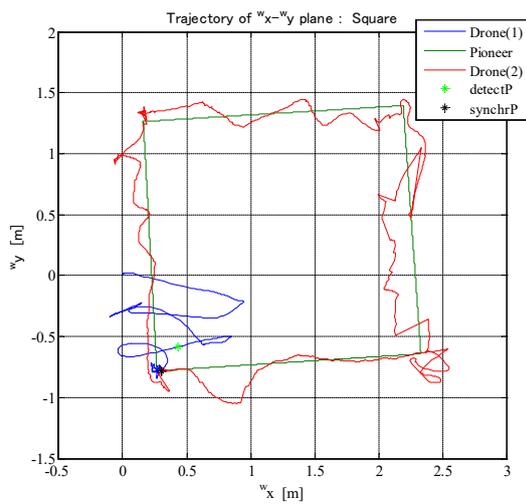


(a) Trajectory

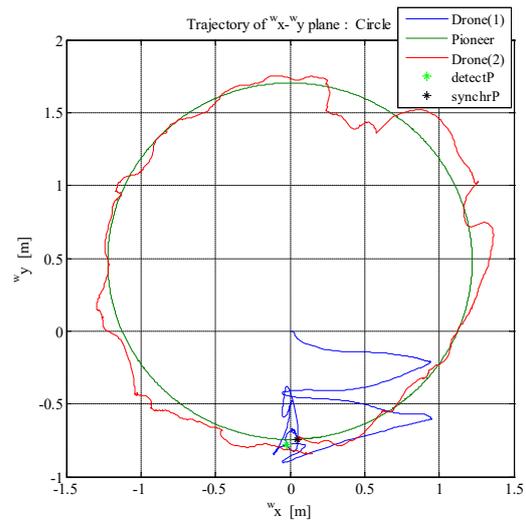


(b) Positional error in formation

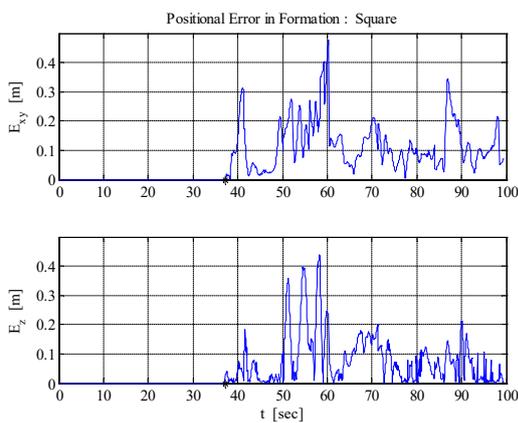
Figure 8: Experimental result of formation control, Line move



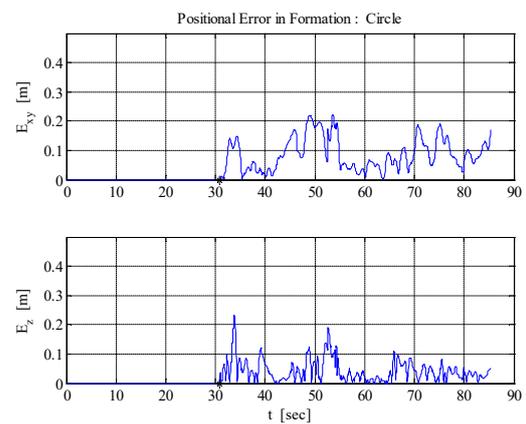
(a) Trajectory



(a) Trajectory



(b) Positional error in formation



(b) Positional error in formation

Figure 9: Experimental result of formation control, Square move

Figure 10: Experimental result of formation control, Circle move

aruco marker on Pioneer was detected at the point “detect” marked by the green asterisk. After hovering over the aruco marker for approximately two seconds, the formation control started; that is, AR.Drone followed Pioneer which moved circularly. Letting  $E_{xy}$  and  $E_z$  be the positional error in the  $w_x$  -  $w_y$  plane and the altitude error, the mean and the maximum are summarized in Table 2.

### V. CONCLUDING REMARKS

This paper has presented an autonomous flight control system of a quadcopter AR.Drone 2.0 only using the internal sensor. Since AR.Drone was designed as a quadcopter with four control commands, four velocity models were constructed by the step response experiments. The time-delay

Table 2: Positional errors in formation control, Unit: (m)

	mean( $E_{xy}$ )	max( $E_{xy}$ )	mean( $E_z$ )	max( $E_z$ )
Line	0.103	0.502	0.095	0.944
Square	0.119	0.478	0.081	0.438
Circle	0.087	0.222	0.043	0.231

due to the communication between AR.Drone and PC and the nonlinearities in the control commands were also included in the models. The basic control laws for tracking and positioning control were combined with each other to construct the complex control system for required flight missions. Furthermore, the proposed autonomous flight control system was applied to a formation control with a

mobile robot. The effectiveness of the proposed autonomous flight control system was demonstrated in experiments.

## REFERENCES

- [1]. K. Nonami, Drone technology, cutting-edge drone business, and future prospects, *Journal of Robotics and Mechatronics*, 28(3), 2016, 262-272.
- [2]. K. Sato and R. Daikoku, A simple autonomous flight control of multicopter using only web camera, *Journal of Robotics and Mechatronics*, 28(3), 2016, 286-294.
- [3]. H. Ukida and M. Miwa, LED panel detection and pattern discrimination using UAV's on-board camera for auto-flight control, *Journal of Robotics and Mechatronics*, 28(3), 2016, 295-303.
- [4]. Y. Morikawa, Developments of autonomous unmanned helicopter and subsequent studies, *Journal of Robotics Society of Japan*, 34(2), 2016, 98-99.
- [5]. R. Lozano, *Unmanned Aerial Vehicles*, (Wiley, 2010).
- [6]. X. Ding, X. Wang, Y. Yu and R. Lozano, Dynamic modelling and trajectory tracking control of a quadrotor unmanned aerial vehicle, *Journal of Dynamic Systems, Measurement, and Control*, 139, 2017, 021004-1-11.
- [7]. A. Aboudonia, A. El-Badawy and R. Rashad, Disturbance observer-based feedback linearization control of an unmanned quadrotor helicopter, *Journal of Systems and Control Engineering*, 230(9), 2016, 877-891.
- [8]. M. Ariffanan, M. Basri, A. R. Husain and K. A. Danapalasingam, A hybrid optimal backstepping and adaptive fuzzy control for autonomous quadrotor helicopter with time-varying disturbance, *Journal of Aerospace Engineering*, 229(12), 2015, 2178-2195.
- [9]. K. Umemoto, T. Ikeda and F. Matsuno, Robust tracking control for multi-rotor UAVs using sliding mode control, *Transactions of the Society of Instrument and Control Engineers*, 50(2), 2014, 170-176.
- [10]. L. Sun and Z. Zuo, Nonlinear adaptive trajectory tracking control for a quad-rotor with parameter uncertainty, *Journal of Aerospace Engineering*, 229(9), 2014, 1709-1721.
- [11]. J. Hamada, S. Suzuki, T. Ichikawa, H. Kurihara and K. Sumida, Adaptive PID control of multi-rotor helicopter, *Journal of the Robotics Society of Japan*, 36(7), 2018, 508-515.
- [12]. R. Sanz, P. Garcia, Q.-C. Zhong and P. Albertos, Robust control of quadrotors based on an uncertainty and disturbance estimator, *Journal of Dynamic Systems, Measurement, and Control*, 138, 2016, 071006-1-8.
- [13]. A. Hori, T. Furui, D. Iwakura and K. Nonami, Auto tuning adaptive I-PD control having an on-line system identification mechanism for industrial multi-rotor helicopters, *Transactions of the Japan Society of Mechanical Engineers*, 82(834), 2016, 1-11.
- [14]. C. Liu, W.-H. Chen and J. Andrew, Piecewise constant model predictive control for autonomous helicopters, *Robotics and Autonomous Systems*, 59, 2011, 571-579.
- [15]. J. Engel, J. Stum and D. Cremers, Scale-aware navigation of a low-cost quadrotor with a monocular camera, *Robotics and Autonomous Systems*, 62, 2014, 1646-1656.
- [16]. H. Kawai, T. Murao, Y. Tsuruo and M. Fujita, Visual motion observer-based pose control with input saturation – Application to small unmanned aerial vehicles –, *Transactions of the Society of Instrument and Control Engineers*, 49(6), 2013, 639-645.
- [17]. P.-J. Bristeau, F. Callou, D. Vissière and N. Petit, The navigation and control technology inside the AR.Drone micro UAV, *Proc. of the 18th IFAC World Congress*, Milano, 2011, 1477-1484.
- [18]. Parrot, AR.Drone 2.0, 2012.
- [19]. I. D. Cowling, O. A. Yakimenko, J. F.W. and A. K. Cooke, A prototype of an autonomous controller for a quadcopter UAV, *Proc of European Control Conference*, 2007
- [20]. L. V. Schmidt, *Introduction to Aircraft Flight Dynamics* (AIAA, Reston, Virginia, 1998).
- [21]. S. M. Shinnars, *Advanced Modern Control System Theory and Design* (John Wiley & Sons, New York, 1998).
- [22]. Y. Ukigai, A. Fujimori and S. Oh-hara, A synthesis of autonomous flight control system for a drone, *Proc of Yamanashi District Conference*, 2017, 21-22.
- [23]. B. D. O. Anderson, and J. B. Moore, *Optimal Control Linear Quadratic Methods* (Prentice-Hall, Englewood Cliffs, 1989).
- [24]. G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library* (O'Reilly Media, Sebastopol, 2008).
- [25]. S. Garrido-Jurado, R. Munoz-Salinas, F. J. Madrid-Cuevas and M. J. Marin-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, *Pattern Recognition*, 47(6), 2014, 2280-2292.

Atsushi Fujimori" Autonomous Flight Control System of Quadcopter Based on Waypoint and its Application to Formation Control with Mobile Robot" International Journal of Engineering Research and Applications (IJERA), vol. 9, no. 10, 2019, pp 26-33