TELKOMNIKA, Vol.13, No.3, September 2015, pp. 930~939
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v13i3.1862

Waypoint Navigation of AR.Drone Quadrotor Using Fuzzy Logic Controller

Veronica Indrawati*[1], Agung Prayitno[2], Thomas Ardi Kusuma[3]
Electrical Engineering Department, University of Surabaya (UBAYA)
Jl. Raya Kalirungkut – Surabaya 60293, East Java – Indonesia, Tel.+62-31-2981157
*Corresponding author, e-mail: veronica@staff.ubaya.ac.id[1], prayitno_agung@staff.ubaya.ac.id[2], blackrogu3@gmail.com[3]

Abstract

In this paper, an AR.Drone is flown autonomously from the initial position (x,y,z) to the desired position called waypoint using Fuzzy Logic Controller (FLC). The FLC consists of three control loops which are pitch control loop, roll control loop and vertical rate control loop. For each control loop, desired position and real position are used as inputs of the FLC, while pitch, roll and vertical rate are used as output respectively. The algorithm is realised in three flight schemes and the navigation data is recorded. The first flight scheme: a desired x-position of AR.Drone will be reached first followed by a desired y-position, and lastly a desired z-position. The second flight scheme: a desired x-position and y-position will be reached simultaneously followed by a desired z-position. The third flight scheme: AR.Drone flies towards to desired position simultaneously. The results show that the AR.Drone can reach the waypoint with the three schemes well. However, the flight scheme straight towards the waypoint with the FLC working simultaneously is the most satisfying one.

Keywords*: waypoint navigation, AR.Drone control, fuzzy logic controller*

1. Introduction

Nowadays, quadrotor is not only used as a hobby, but it has also been widely used for various activities, such as news coverage in the affected areas, traffic coverage, the shooting of a region, promotional events and several other shows. Generally, this quadrotor is still flown manually by using remote control. Research at the university has developed a wide range of controllers that can fly quadrotor automatically. Various kinds of controllers have been designed, among others for tracking an object, flying through obstacles, determining formation-flight and tracking the trajectory. The development of the various algorithms is one of most interesting fields of research in most of the leading universities worldwide. The development will be faster if the quadrotor is ready in hardware. One of the most commonly used quadrotors is the AR.Drone.

AR.Drone is a quadrotor made by Parrot, a French company. At first, the AR.Drone is made as a toy for the sake of entertainment, which can be played with applications installed in Android devices and iOS devices through Wi-Fi. AR.Drone has already had several sensors, such as: 3 axis accelerometer, 3 axis gyroscope, a sonar altimeter, and the front and bottom cameras. Moreover, this drone is equipped with an onboard computer that can be used for vertical take off, landing, hovering, and video streaming from two cameras via Wi-Fi [1].

Parrot has also released an official SDK [2] that can help users to access the innerboard of the AR.Drone. When the AR.Drone is turned on, the innerboard will automatically act as a server which is complemented by the facilities of Dynamic Host Configuration Protocol (DHCP), so that users can connect to the AR.Drone without having to set up an Internet Protocol (IP) on their computers. By using the innerboard, users can control the main flight (take-off, hovering, landing, and emergency stop) and manoeuvre the flight by giving value within the range of -1 to 1 in the pitch, roll, yaw rate and vertical rate input. A value of -1 and 1 will represent the minimum and maximum value of each input whose value can be set from the innerboard configuration. The value indicates the angels pitch angle, roll, yaw rate and vertical rate that are proportional with the minimum and maximum range. Positive and negative values indicate the directions. Positive values (+) in the pitch cause the drone to move backward, while negative

Received April 15, 2015; Revised July 9, 2015; Accepted July 22, 2015

values (-) cause the drone to fly forward. To manoeuvre to the right, roll input is given a positive value, while to the left means giving the input roll a negative value. To manoeuvre the pivot clockwise, yaw rate input is given a positive value, and counter-clockwise means giving the input yaw rate a negative value. To fly up, the vertical rate is given a positive value, while to move down, the vertical rate is given a negative value. The point is this, that to control the AR. Drone is to send commands to the innerboard and receive navigation data (NavData) from the innerbaord via Wi-Fi. These commands are in the form of pitch, roll, yaw rate and vertical rate, while the Nav–Datas are in the form of actual pitch value, forward speed, actual roll value, sideward speed, actual yaw rate value, yaw value, vertical rate value and altitude value

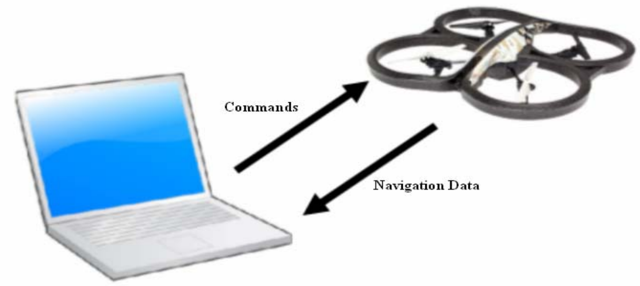


Figure 1. AR.Drone send commands and receive the navigation data

Based on this description, the AR.Drone is chosen to be the platform of this research. The type of AR.Drone used in this research is the AR.Drone 2.0 Elite Edition which has the following specifications: 4 inrunner brushless motors. 14.5W 28,500 RPM, 32-bit ARM Cortex A8 1GHz processor with 800MHz DSP TMS320DMC64x video, 1GB DDR2 RAM at 200MHz, 3-axis gyroscopes 2000°/second precision, three axis accelerometers +/- 50 mg precision, three-axis magnetometers 6° precision, Pressure +/- 10 Pa precision sensors, ultrasound sensors for measurement of ground altitude, 60 FPS QVGA vertical ground speed cameras for measurement, Linux 2.6.32, USB 2.0 high speed for extensions, Wi-Fi, HD Camera. 720p 30fps [3].

Several studies using the AR.Drone as a platform among others are described in this section. Pierre-Jean Bristeau et al. [4] explained in detail that the navigation technology and control used in the AR.Drone include the hardware description, vision algorithm, sensor calibration, altitude estimation, velocity estimation and control architecture. Nick et al. [5] made an AR.Drone simulation with the sensor and motion models. They also made a visual map and indoor environment. Using the visual map, the AR.drone can localise itself. Michael Mogenson [1] made an AR.Drone LabVIEW toolkit to control the AR.Drone 1.0. Broadly speaking, this software consists of several Virtual Instruments (VI) which are the main VI, video VI, NavData VI, supporting VI’s and additional VI’s. This software is made to make it easier for researchers and teachers to learn about AR.Drone. Krajnik et al. [6] created a model structure of the AR.Drone which consists of 4 models: pitch, roll, yaw rate and vertical rate. The model parameter is earned from the estimation result using the data from the experiment. Agung et al. [7] implemented fuzzy logic controller in the AR.Drone 2.0 for the trajectory tracking application. Some forms of trajectory tracking have been successfully followed by the drone. Sarah Yifang [8] obtained the dynamics model of AR.Drone that consists of internal controller model and the physical dynamics of the drone. Some controller algorithm is applied to the drone, such as waypoint navigation and trajectory, following with PID controller and also vision-based controller for a variety of flight formation. Rabah Abbas et al. [9] proposed a PID controller and directed lyapunov controller for formation tracking of quadrotors. PID controller is implemented on the leader quadrotor, while directed Lyapunov controller on the followers. Dynamics optimisation of the parameters controllers is achieved using an artificial fish swarm algorithm. Emad Abbasi et al. [10] simulated two control schemes to control the height of the quadrotor. The first scheme uses 4 PID controllers which are then simulated using turbulence signal. The second scheme uses combination fuzzyPID controller which are also simulated using the same turbulence signal. The result of the simulation shows that the fuzzyPID combination is more suitable with the turbulence situation. Abbasi et al. [11] compared the classical PID controller and the fuzzy supervisory controller for tuning the PID controller to stabilise the quadrotor modelled with Euler-Newton equation. The result of the simulation shows that fuzzyPID is better than PID in the case of eliminating overshoot and shortening the settling time. Santos et al. [12] simulated fuzzy logic to control the model of the quadrotor. The input is the height, roll, pitch and yaw value; the output is the power of each of the four rotors. The result of the simulation shows the efficiency from the control strategy. Senthil Kumar et al. [13] simulated fuzzy logic to control the model of the quadrotor using Fuzzy Logic Toolbox Matlab. The fuzzy used has 3 inputs, which are error (the difference between the desired value and the present value), derivative error and integral error. The output is the control value power of each motor.

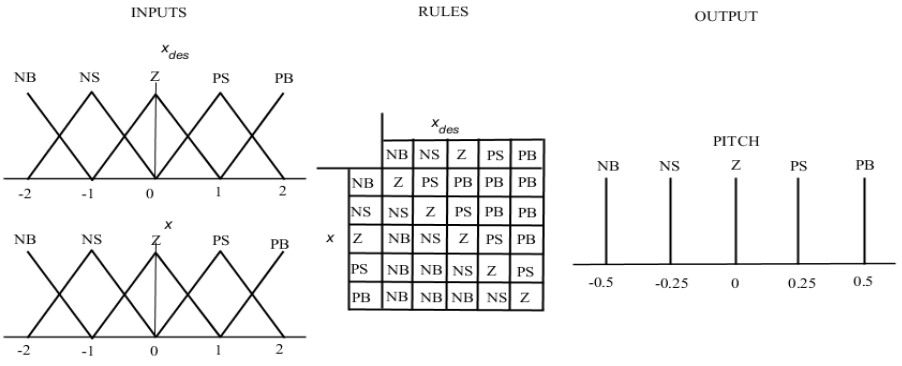
Waypoint navigation is a new technology that allows for the drones to fly from one point to another. With this technology, the drones can fly at a certain height, at a certain speed, with certain fly patterns and hover at the destination point with the remote control navigation software. In the future this technology will be very helpful, especially for business and social missions. For example, it can be used in the delivery of goods for business or humanitarian missions in disaster areas. This technology typically utilises GPS and a map on the computer screen for monitoring and control.

In this paper, waypoint navigation technology will be implemented in the laboratory using AR.Drone as a platform. The AR.Drone will be designed to fly from the initial position *(x,y,z)* to a desired waypoint *(x_{des},y_{des},z_{des})* with various schemes. The algorithm of the fuzzy logic controller will be used for remote control navigation which is realised using LabVIEW software. The implementation of the fuzzy algorithm used for controlling the AR.Drone has not yet been done by many researchers. Therefore, the fuzzy control scheme for the waypoint application will provide benefits for the development of AR.Drone control.

2. Research Method

In the research, three schemes of waypoint navigation AR.Drone will be implemented using the fuzzy logic controller made by the LabVIEW software. Waypoint navigation is a flying command of the AR.Drone from its initial position (*x,y,z*) to the desired position (x_{des},y_{des},z_{des}) which is known as the waypoint. For the flying manoeuvring, we use three control signals, which are pitch, roll and vertical rate and are the results of three fuzzy logic controllers. The design of fuzzy logic control in this study considers the following points. The field used is 4 metres in length, 4 metres in width and 4 metres in height. Assuming the initial position whilst flying is in the centre of the field, the range of the fuzzification input position and reference position is between -2 to 2 metres. The range of each output is based on an empirical method to determine the value range so that the speed is not too slow or too fast. Singleton is chosen for its speed in calculating the defuzzification process. The details of each controller are described below:

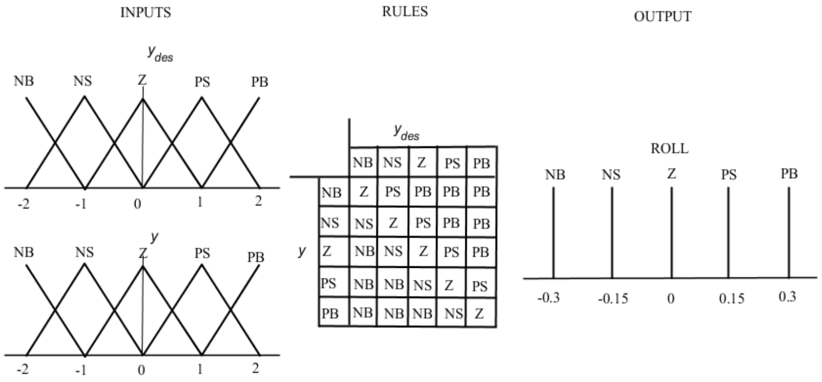
a) To reach the coordinate of the desired *x*-position (x_{des}), a pitch control loop and two fuzzy inputs are designed, which are the desired coordinates of *x* and the *x*-position from the NavData. The range of the fuzzification is -2 to 2 metres, which is stated in the 5 triangular membership functions. Meanwhile, the fuzzy output is the pitch value in the range of -0.5 to 0.5 which is stated in 5 singletons. Further details of this design are shown in Figure 2.



x \ x_{des}	NB	NS	Z	PS	PB
NB	Z	PS	PB	PB	PB
NS	NS	Z	PS	PB	PB
Z	NB	NS	Z	PS	PB
PS	NB	NB	NS	Z	PS
PB	NB	NB	NB	NS	Z

Figure 2. The fuzzy control of pitch

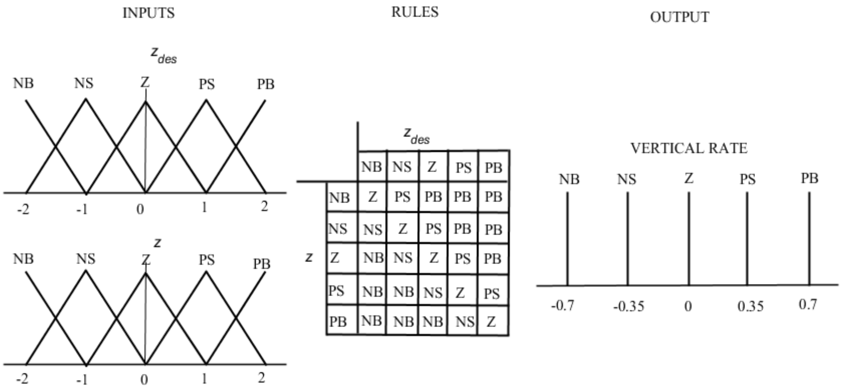
b) To reach the coordinate of the desired *y*-position (y_{des}), a roll control loop and two fuzzy inputs are designed, which are the desired coordinates of *y* and the *y*-position from the NavData. The range of the fuzzification is -2 to 2 metres, which is stated in 5 triangular membership functions. Meanwhile, the fuzzy output is the roll value in the range of -0.3 to 0.3 which is stated in 5 singletons. The details of the design are shown in Figure 3.



y \ y_{des}	NB	NS	Z	PS	PB
NB	Z	PS	PB	PB	PB
NS	NS	Z	PS	PB	PB
Z	NB	NS	Z	PS	PB
PS	NB	NB	NS	Z	PS
PB	NB	NB	NB	NS	Z

Figure 3. The fuzzy control of roll

c) To reach the coordinate of the desired *z*-position (z_{des}), a vertical control rate and two fuzzy inputs are designed, which are the desired coordinates of *z* and the *z*-position from the NavData. The range of the fuzzification is -2 to 2 metres, which is stated in 5 triangular membership functions. Meanwhile, the fuzzy output is the vertical rate value in the range of -0.7 to 0.7 which is stated in 5 singletons. Further details of this design are shown in Figure 4.



z \ z_{des}	NB	NS	Z	PS	PB
NB	Z	PS	PB	PB	PB
NS	NS	Z	PS	PB	PB
Z	NB	NS	Z	PS	PB
PS	NB	NB	NS	Z	PS
PB	NB	NB	NB	NS	Z

Figure 4. The fuzzy control of vertical rate

Using the three FLCs, the waypoint is obtained with three flight schemes, which are reaching the waypoint in three sequences, reaching waypoint in two sequences and reaching waypoint in one sequence. Surely for each of these flight schemes the coordinates of each FLC is needed and explained below.

2.1. Reach Waypoint in Three Sequences

In this scheme, the AR.Drone will reach the desired waypoint coordinate (x_{des},y_{des},z_{des}) by reaching the desired *x*-position (x_{des}) first, followed by the desired y-position (y_{des}), and lastly the desired *z*-position (z_{des}). The three FLCs (pitch, roll, vertical rate) will work together in dependence. The controller in the flight scheme works this way:

a) The FLC pitch system makes the AR.Drone move towards the x_{des} coordinate, and shuts down the FLC roll and FLC vertical rate system. When the AR.Drone reaches x_{des}, the pitch FLC system will send logic signals to activate the roll FLC system.
b) The activation of this FLC roll system makes the AR.Drone move towards the y_{des} position and stops the FLC pitch and vertical rate system. When the AR.Drone reaches y_{des}, the FLC roll system will send logic signals to activate the vertical rate FLC system.
c) The activation of this FLC vertical rate system makes the AR.Drone move towards the z_{des} position and stops the FLC pitch and roll system. When the AR.Drone reaches z_{des}, the FLC roll system will send logic signals to activate the FLC pitch system back and repeats the sequence above.
d) This process is done because while switching the FLC, a change in the position may occur. In order that the AR.Drone is always on waypoint (x_{des},y_{des},z_{des}), the FLC must be conducted using the sequence above so that it can hover.

The orders of flight and diagram blocks of the controlled system to finish the flight scheme are shown in Figure 5 and 6.

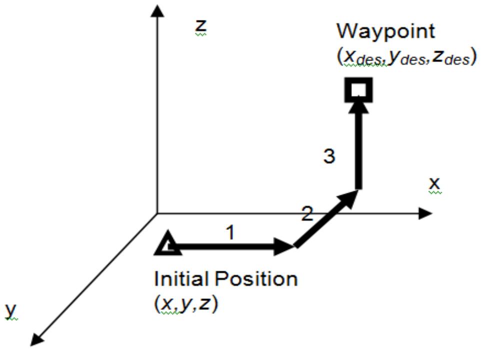


Figure 5. Scheme of reach waypoint in three sequences

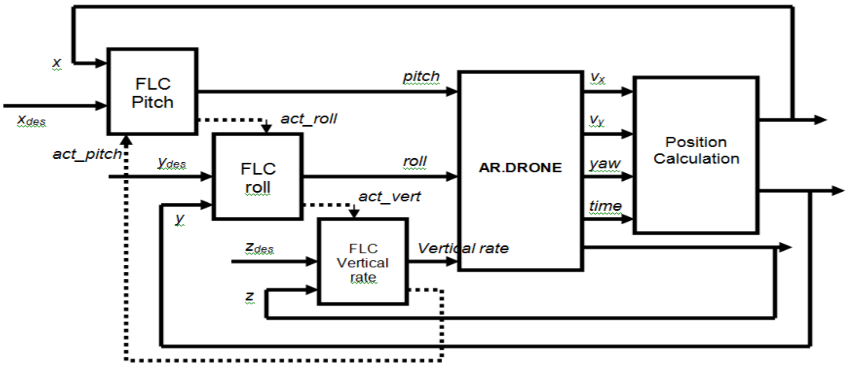


Figure 6. Control architecture of the scheme of reach waypoint in three sequences

2.2. Reach Waypoint in Two Sequences

In this scheme, the AR.Drone will reach the desired coordinate of the waypoint (x_{des},y_{des},z_{des}) with flying towards the (x_{des},y_{des}) coordinate first and then flying towards the desired z-position (z_{des}). Three FLCs (pitch, roll, vertical rate) work this way:

a) The FLC pitch and roll system will be turned on at the same time so that the AR.Drone moves toward the x-y field and to the (x_{des},y_{des}) coordinate directly. After that, the two FLC systems will send logic signals to activate the FLC vertical rate system.
b) Once the FLC vertical rate system is activated, the AR.Drone will move towards the z_{des} coordinate and stop the FLC pitch and roll system. After reaching the z_{des}, the vertical rate system will send logical signals to activate the FLC pitch and roll system back.

c) The position of the AR.Drone on waypoint (x_{des},y_{des},z_{des}) should always be maintained using the control sequence above.

The orders of flight and diagram blocks of the controlled system to finish the flight scheme are shown in Figure 7 and 8.

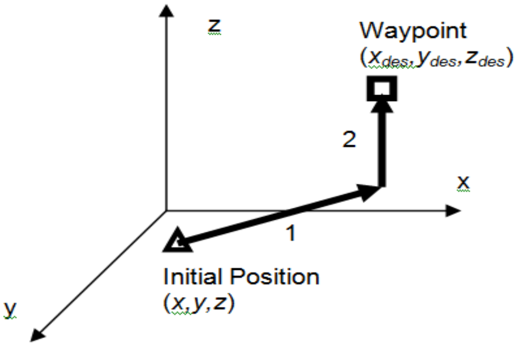


Figure 7. Scheme of reach waypoint in two sequences

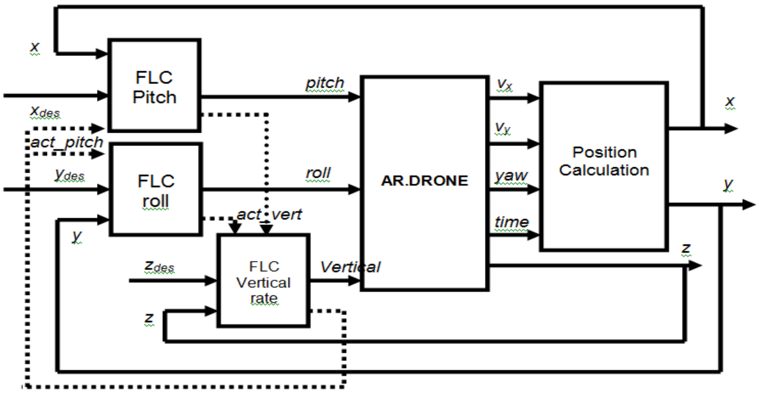


Figure 8. The control architecture of scheme of reach waypoint in two sequences

2.3. Reach Waypoint in One Sequence

In this scheme, the AR.Drone will reach the desired waypoint coordinate (x_{des},y_{des},z_{des}) by flying directly towards those coordinates. Three FLCs (pitch, roll, vertical rate) will work simultaneously and each will be responsible for its position.

The orders of flight and diagram blocks of the controlled system to finish the flight scheme are shown in Figure 9 and 10.

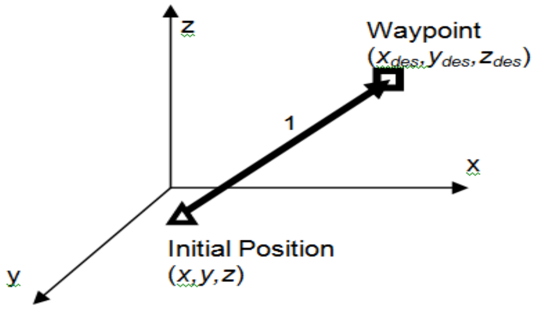


Figure 9. Scheme of reach waypoint in one sequence

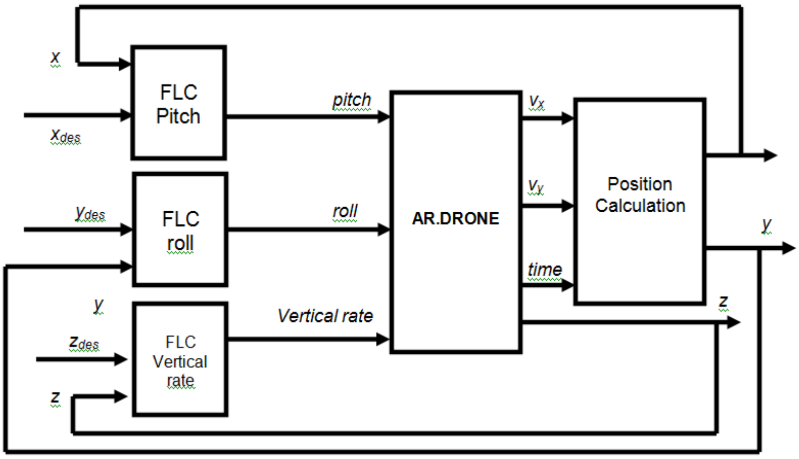


Figure 10. The control architecture of scheme of reach waypoint in one sequence

To implement the flight schemes above, several subVI and front panels in the LabVIEW software are made. Several main subVI, such as the subVI used for flying and subVI used to read the NavData, modify the subVI in the AR.Drone LabVIEW toolkit which was made by Michael Mogenson [1, 14] for AR.Drone 1.0 so that it could be used for the AR.Drone 2.0. The inputs of this AR.Drone system are the pitch value, roll, yaw rate and vertical rate whose values are in the range of -1 to 1. Meanwhile, the variables that could be taken from the AR.Drone are actual pitch value, forward speed, actual roll, sideward speed, actual yaw rate, yaw, vertical rate and altitude. To obtain the positions of *x* and *y*, the subVI position estimation is made. The inputs from the block position estimation are the forward speed (v_x), sideward speed (v_y) and time stamp (*t*). The equation of this estimation of *x* and *y* position is stated as in Equation (1) and (2) below:

$$x_n = x_{n-1} + v_{xn}(t_n - t_{n-1}) \quad (1)$$

$$y_n = y_{n-1} + v_{yn}(t_n - t_{n-1}) \quad (2)$$

Whereas *n* is the present sample data and *z* position is the direct result of the ultrasonic sensor onboard. These equations result in the subVI position estimation. The FLC block is realised into the subVI Fuzzification, subVI Inference, and subVI Defuzzification.

3. Results and Analysis

The algorithm of the FLC is implemented in the AR.Drone, which is flown autonomously in a closed space using LabVIEW, Figure 11. The procedures for testing are:

a) Through the front panel software, the AR.Drone is flown in hover mode 1 metre from the ground. That point is called the initial position with the coordinate value (0,0,1).
b) Next, the desired waypoint coordinate is inserted through the front panel. By switching off the hovering mode, the AR.Drone will fly autonomously with the made FLC control towards the waypoint spot.
c) While flying from the initial position to the waypoint coordinate, the actual *x*-position, *y*-position and *z*-position values are recorded.
d) After reaching the waypoint coordinate, indicated with hover mode, the AR.Drone will be landed towards the ground station.

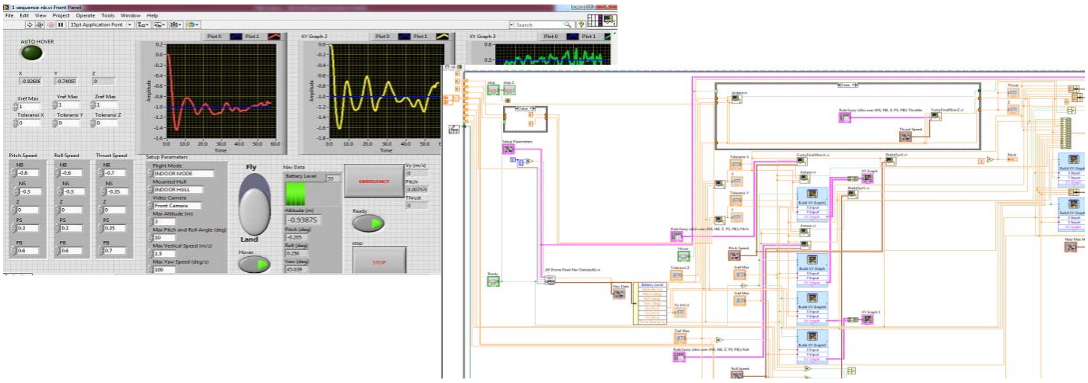


Figure 11. The front panel and block diagram of waypoint navigation

The result of the FLC algorithm for the flight scheme “reach waypoint in three sequences” is shown in Figure 12.

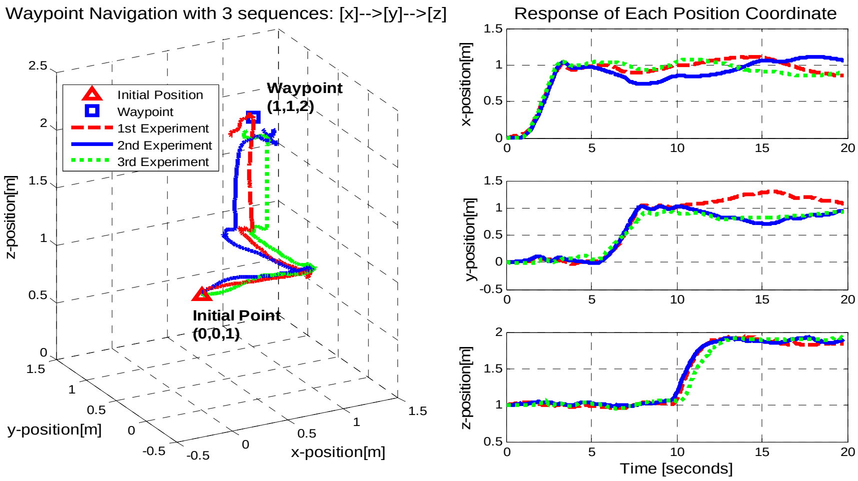


Figure 12. The experiment results of scheme reach waypoint in three sequences

Figure 12 shows the results of the experiment done three times, from the initial position towards the waypoint. Generally, the AR.Drone can do control commands made for it to fly towards the *x*-position first, followed by the *y*-position, and to the height of the desired *z*-position. It can be seen that when the *x*-position is reached and it is moving towards the *y*-position, there is a shift of the *x*-position away from the setpoint. Exactly the same thing happens when the *y*-position is reached and the drone is moving towards the *z*-position. This happens because of the switching enable process and disablement of the three FLCs that are being used. The problem can also occur because the values of the x and y positions are the result of the estimated output of the block position calculation, not the sensor readings directly. However, generally each position can be reached at around 4 seconds while the waypoint is reached at around 15 seconds.

The next testing is for the flight scheme “Reach Waypoint in Two Sequences”, which was also done three times. The result is shown in Figure 13.

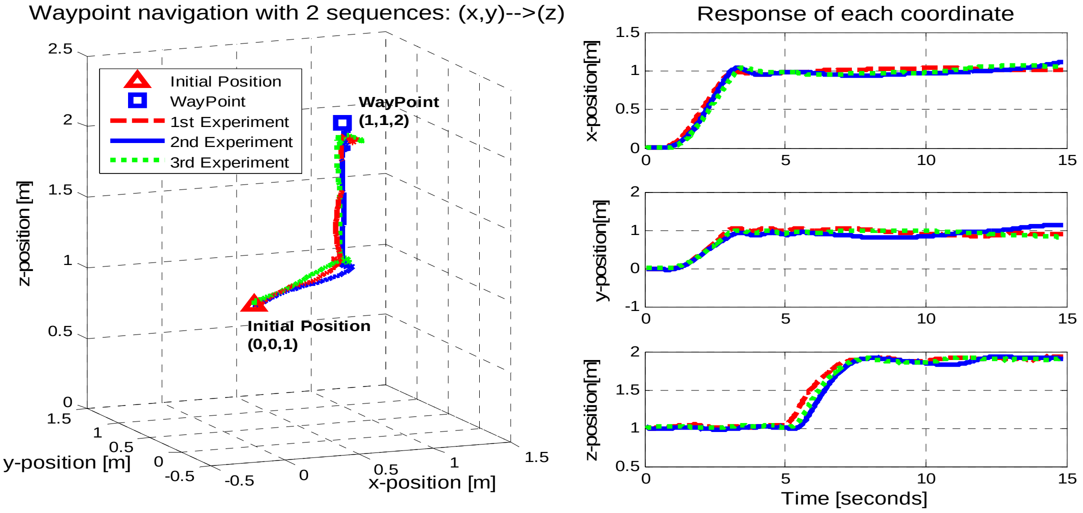


Figure 13. The experiment results of scheme reach waypoint in two sequences

The result shows that the positions (x_{des},y_{des}), can be reached well simultaneously, but when it moves towards the z_{des}, the shift of the x_{des} and y_{des} from the setpoint can be seen. This also happens because the control pitch and roll switches off when the vertical rate control is working. Again, the effects of the estimated positions x and y are still visible. The waypoint can be reached at around 9 seconds.

The last testing is done for the flight scheme “Reach Waypoint in One Sequence”, where the AR.Drone flies towards the waypoint (x_{des},y_{des},z_{des}) directly. The result is shown in Figure 14.

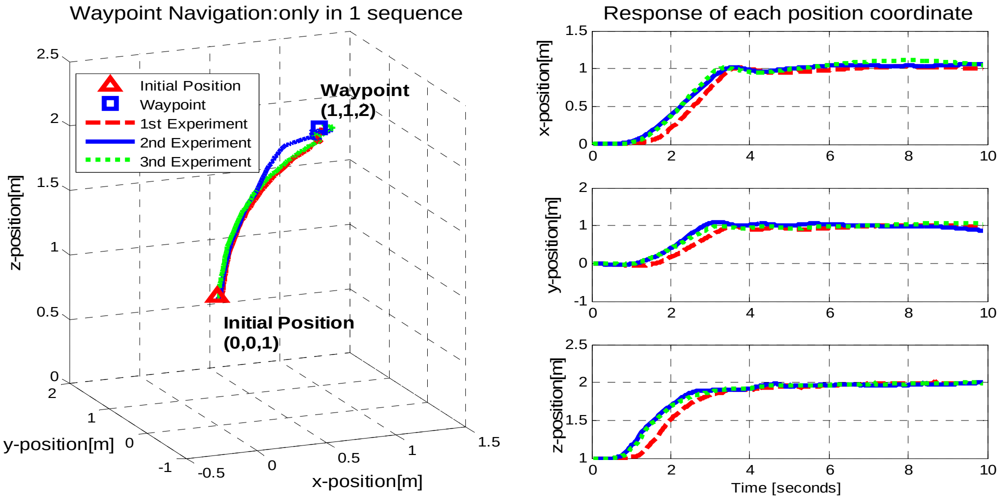


Figure 14. The experiment results of scheme reach waypoint in one sequence

The results of the experiments (done 3 times) show that the AR.Drone can reach waypoint (x_{des}, y_{des},z_{des}) with the settling time less than 4 seconds. The response when it is steady shows a relatively better result than the two previous flight schemes. This is caused by the 3 FLCs working simultaneously.

4. Conclusion

Generally, the three flight schemes can be implemented using three FLCs (FLC pitch, FLC roll, FLC vertical rate) for the waypoint navigation. The results of these tests show that the flight scheme straight towards the waypoint with the FLC working simultaneously is the most satisfying compared to the other two flight schemes. Calculation of the positions (*x* and *y*) is still susceptible to noise. Use can be made of a compensator on the side of the pitch and roll to get better results.

References

[1] Michael M. The AR.Drone LabVIEW Toolkit: A Software Framework for the Control of Low Cost Quadrotor Aerial Robots. Master of Science Thesis. TUFTS University; 2012.
[2] Stephane P, Nicolas B. AR.Drone Developer Guide. *Parrot. SDK 1.6*. 2011.
[3] http://ardrone2.parrot.com accessed on 11 August 2014.
[4] Pierre-Jean B, Francois C, David V, Nicolas P. *The Navigation and Control Technology Inside the AR.Drone Micro UAV*. 18th IFAC World Congress. Milano, Italy. 2011.
[5] Nick Dijkshoorn, Arnoud Visser. Integrating Sensor and Motion Models to Localize an Autonomous AR.Drone. *International Journal of Micro Air Vehicle.* 2011; 3(4): 183-200.
[6] Krajnik T, Vonasek V, Fiser D, Faigl J. *AR-Drone as a Platform for Robotic Research and Education*. Research and Eductation in Robotics: EUROBOT. Heidelberg. 2011.
[7] Agung Prayitno, Veronica Indrawati, Gabriel Utomo. Trajectory Tracking of AR.Drone Quadrotor Using Fuzzy Logic Controller. *TELKOMNIKA Telecommunication Computing Electronics and Control.* 2014; 12(4): 819-828.
[8] Sarah Yifang Tang. Vision-Based Control for Autonomous Quadrotor. Final Report: Undergraduated Senior Thesis. Department of Mechanical and Aerospace Engineering, Princeton University; 2013.
[9] Rabah Abbas, Qinghe Wu. Improved Leader Follower Formation Control for Multiple Quadrotors Based AFSA. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2015; 13(1): 85-92.
[10] Emad Abbasi Seidabad, Saeed Vandaki, Ali Vahidin Kamyad. Designing Fuzzy PID Controller for Quadrotor. *International Journal of Advanced Research in Computer Science & Technology (IJARCST)*. 2014; 2(4): 221-227.
[11] E Abbasi, MJ Mahjoob. Controlling of Quadrotor UAV Using a Fuzzy System for Tuning the PID Gains in Hovering Mode. 2015.
[12] Matilde Santos, Victoria Lopez, Franciso Morata. *Intelligent Fuzzy Controller of a Quadrotor*. IEEE Intelligent Systems and Knowledge Engineering Conf (ISKE). 2010.
[13] K Senthil Kumar, Mohammad Rasheed, R Muthu Madhava Kumar. *Design and Implementation of Fuzzy Logic Controller for Quad Rotor UAV*. 2nd International Conference on Research in Science, Engineering and Technology (ICRSET'2014). Dubai. 2014.
[14] https://ardronelabviewtoolkit.wordpress.com