ISSN: 0974-5823 Vol. 7 No. 1 January, 2022

International Journal of Mechanical Engineering

A STUDY ON EMBEDDED SYSTEM FOR AUTONOMOUS FLIGHT AND COLLISION AVOIDANCE OF QUAD-ROTOR USING DEVELOPMENT ENVIRONMENT

[1]Seongse Cho, [2]Hye-Gi Park and [3]Wonhyuk Choi*

[1,2]Student, Dept. of Aeronautical Systems Engineering, Hanseo University, Taean 32158, Korea

[3*]Professor, Dept. of Marine sports, Hanseo University, Taean 32158, Korea

ABSTRACT

MAV (Micro Air Vehicles) is a research field that is attracting attention due to its small size and various applicability among remote-controlled UAVs (Unmanned Aerial Vehicles). In particular, Quad-Rotor is being targeted for autonomous flight missions suitable for various situations and purposes due to its ease of design and flight control, and various applications ranging from military and civilian commercial and personal use are being developed. Meanwhile, ROS (Robot Operating System) first developed in 2008 and used for the development of robot control software is a Meta operating system that provides various libraries and examples for robot control system development, as well as SITL and HITL simulation environments for development algorithms. As a result, developers have a lot of benefits in terms of reducing costs, time, and risk. This paper proposes an autonomous flight and collision avoidance algorithm of a 450-type Quad Rotor (PX4) assembled with commercial parts using the ROS environment in the process of developing an embedded system for obstacle detection and collision avoidance of MAV. In addition, this paper was written for the purpose of demonstrating its usefulness through simulation and flight test by implementing an embedded system mounted on an aircraft to which it is applied.

Keywords: *MAV, Embedded System, Autonomous Flight, ROS, Quad Rotor, Collision Avoidance*

1 INTRODUCTION

Recently, as the field of use of MAV has been expanded in modern society, research on MAV flight control is receiving more attention. Diversity in its form and application fields, such as terrain modeling [1], surveillance of civilian protection facilities [2], aerial surveillance [3] for military purposes, and utilization of delivery operations for commercial purposes [4], are gradually expanding. Among the related studies, the number of proposals related to autonomous flight that can follow a number of predetermined route points and perform missions without human intervention in the middle has increased [5]. ROS [6] is used by many researchers as a development environment because it provides a convenient development environment to test structural functions in advance and monitor communication between modules in the process of developing algorithms responsible for flight control and path planning of these MAVs. In particular, the global community of developers constitutes a vast ecosystem of open-source data, providing the advantage of achieving optimal results without trial and error even for beginners who are unfamiliar with development. Among the many packages in the ROS community, MAVROS is a specialized package for MAV. MAVROS is installed on aircraft-mounted systems and is modularized to enable interfaces between ROS and flight control modules, embedded systems, and remote control systems (mission planners) using a communication protocol for MAVlink.

This study implements a framework for performing integrated systems and autonomous flight missions with Flight Controller with PX4 Firmware, Astra's IR Depth Camera obstacle detection/avoidance image sensor, and a Jetson xavier embedded system with a Ubuntu-based ROS environment. The integrated embedded system processes data in real time and allows the mission to be carried out along a pre-planned path through the mission planner QGroundControl.

In this paper, Chapter 2 introduces recent research trends related to autonomous flight of MAVs. Chapter 3 describes the overall system configuration of MAV, such as path control. Chapter 4 presents the results of the SITL, HITL simulation flight test in the ROS environment. Chapter 5 concludes by presenting conclusions and future research directions.

2 Related Research Trends

Any studies have been conducted on autonomous flight, including path-following and collision avoidance, while excluding the intervention of remote pilots during the MAV's mission. To this end, the MAV is equipped with sensors of various characteristics, and information on the surrounding environment received through signal media such as image, radio, sound, and infrared is mathematically modeled considering the physical characteristics of the MAV. In addition, Kalman filter, Comprehensive filter, etc., which can be estimated by stochastic calculations while requiring relatively low calculation costs despite external interference and confusion, enable accurate measurement of the position and posture of the aircraft, and increase reliability in remote control. However, the reality is that most models still remain within the constraints of communication distance, assuming real-time remote control flight from the ground [7].

Pixhawk, an open hardware platform developed for MAV, has the advantage of having a wide user and developer ecosystem and providing communication compatibility between various hardware and software, making it easier to develop application control systems and interface of mounting sensors. In addition, PX4 Firmware can easily integrate with Ubuntu of embedded systems mounted on Pixhawk, and these advantages are easy to configure multi-threaded embedded systems developed for robotic applications [8].

The ROS architecture is now an integrated robot development tool. Various applications have recently been proposed in consideration of their compatibility with Pixhawk via MAVROS, for example, the extension of the operational concept of increased autonomy in multi-mission performance, including reconnaissance and surveillance, autonomous landing, and cooperation between ground bases and aircraft [9].

There is a Vector Field Histogram (VFH) method that has been used for a long time as an autonomous image-based system for environmental mapping and exploration related to obstacle avoidance/collision prevention in MAV. The first proposed algorithm in 1989 by Johann Borenstein was the existence of a field with forces between an airplane, an obstacle, and a destination, consisting of a Vector with forces acting on a push force and a force acting on a destination [10]. Subsequently, algorithms supplementing each step's limitations such as VFH, VFH+ and VFH* were proposed, and in 2014, 3DVFH+ considering stereoscopic maneuver in space was proposed by Simon Vanneste and others [11].

The conclusions obtained from these studies show that it is possible to implement a lightweight platform with high computational power for the development of autonomous flight capabilities of MAVs through real-time data processing without relying on remote ground station communication links.

3 System Configuration

The hardware configuration of the integrated embedded systems proposed in this paper is shown in Figure 1.

Mounted on a 450mm quadrotor assembled using COTS components, it is tasked to detect/avoid obstacles and reach the destination while following a given path without relying on remote control by pre-planning.

The Depth Camera, mounted in front of the aircraft, provides the embedded system with Point Cloud Data and stereo images for obstacles within a detection distance (0.2–8 m) in an IR pattern.

Embedded computers calculate the probability of the presence of obstacles on the path from image and Point Cloud Data, position of obstacles, posture, movement direction and speed, and provide the flight control module with commands for optimal path tracking based on the current flight path direction, position, posture, and speed.

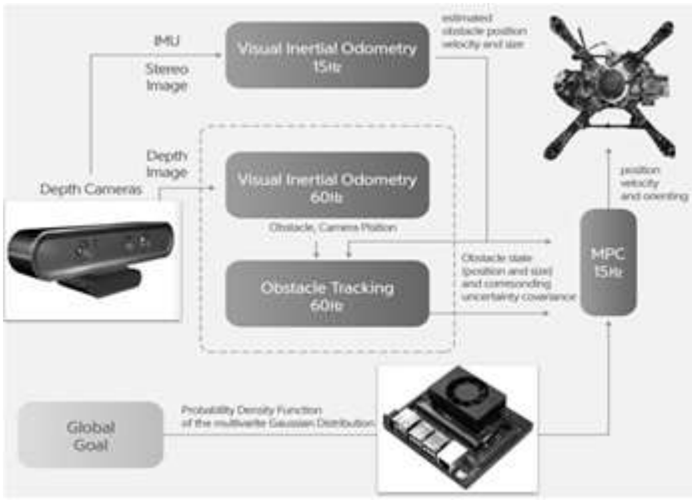

Fig. 1: System Hardware Configuration

The operator pre-defines the mission through the route points it takes. Then, the file containing the set path point is uploaded to the embedded system. When power is applied to the vehicle, the embedded computer algorithm checks whether or not arming is present, autonomously executes the PX4 control command, enters the system into autonomous control state, and autonomously performs the mission following the input path point.

3.1 Obstacle Detection/Collision Avoidance Algorithm Configuration

In this paper, we applied the VFH+ algorithm for obstacle detection and collision avoidance and used a method to optimize parameters additionally.

Step 1 generates an early-stage Pola Histogram in the direction of progress based on Point Cloud Data and calculates the safe passage angle range (S_{max}) by reflecting the size of the aircraft in Point Cloud Data location and distance.

Step 2 produces a Binary Pola Histogram with binaries of passability considering the safety angle for early-stage polar coordinate histograms.

Step 3 identifies the Masked Polar Histogram to determine the extent of obstacle avoidance considering speed and minimum turning radius, and whether it is possible to proceed by azimuth, considering the physical characteristics of MAV.

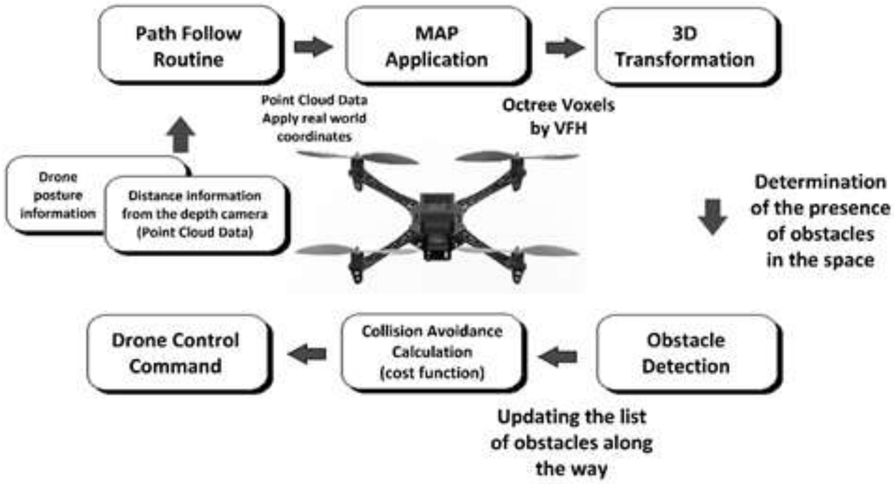

Fig. 2: Functional Block Diagram

In the final stage, the Masked Pola Histogram is used to search the passable empty space (C_l, C_r) on the left and right sides to determine the optimal direction of progress with the evaluation function g(c).

In the case of empty spaces (C_l, C_r) > Smax, as shown in Figure 3, Equation (1) is shown. Empty space (C_l, C_r) < Smax, as shown in Equation (2). where k_l and k_r are the left/right boundaries of the entire space.

$$C_l = k_r + \frac{S_{max}}{2}, C_r = k_l + \frac{S_{max}}{2} \quad (1)$$

$$C_l = C_r = \frac{k_r + k_l}{2} \quad (2)$$

The evaluation function G(c) is composed as shown in Equation (3).

$$g(c) = \mu_1 \Delta(c, \theta_t) + \mu_2 \Delta\left(c, \frac{\theta_i}{\alpha}\right) + \mu_3 \Delta(c, \theta_{n,i-1}), C \in [C_l, C_r] \quad (3)$$

Each term in order means the destination direction θ_t, the current direction θ_i, and the choice direction θ_{i-1} of the previous state, and uses a delta function to calculate the number of sectors between the two zones. The smaller the number of sectors, the less the need for steering from the current direction. In this case, the constant values of μ_1, μ_2, and μ_3 shall be set to '$\mu_1 > \mu_2 + \mu_3$' to increase the weight of the destination.

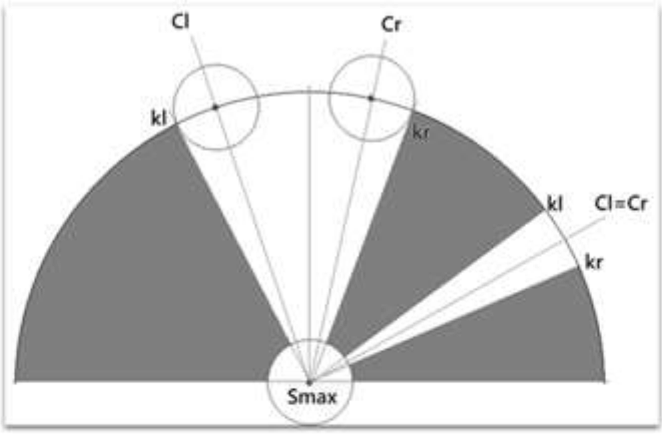

Fig. 3: Decision of direction to proceed

3.2 Framework Development

The autonomous flight integrated embedding system proposed in this paper is implemented as C++ through the MAVROS package provided by ROS. The program module consists of the following executable nodes: controller_node, public_node, verify_node, and so on. PX4 Firmware was built into Pixhawk and Jetson configured the ROS noetic environment based on the LUbuntu operating system. 'controller_node' is the primary control code for the application, which initially checks the input of mission data inside the memory card device and reads the mission path point data in the txt extension file into the system.

After the task is obtained from the txt file, the framework code checks the status of the battery, and if there is no problem, the mission readiness is verified, and the control code starts the service in order in the MAVROS package.

First, a topic '/cmd/arming' is executed to power the UAV motor with a minimum angular speed rotation followed by an offboard flight mode enabled via the '/cmd/guided_enabled' topic. At this point, the manual radio controller for remote control is disabled, and the subsequent mission consists of the ROS message model '/geometry_msgs/PoseStamped'. The ROS message consists of a header, a three-dimensional position value describing the position and posture of the aircraft, and a quaternion to the direction. The obstacle detection/collision avoidance algorithm continuously updates the aircraft's progress commands and warns that the main LED should be re-planned if the residual distance during the mission is greater than the battery's safe flight remaining distance.

The mission status is verified by the obstacle detection/collision avoidance algorithm for each flight path step and the communication message is posted as topic '/SetpointPose_UAV' for Public_node. The controller topic receives feedback from '/local_position/local' and compares the current 3D position of the aircraft with the mission path point.

If the position error between the position of the aircraft and the current path point is in an acceptable area according to the predefined allowable threshold, the current path point is considered to have been reached and control proceeds to the next path point on the mission point list. This is also calculated for the orientation of the aircraft, in which the data are converted from the quaternion to the Euler angle for threshold verification.

A summary of all the steps described above is shown in Figure 4. The Public_node maintains the minimum communication cycle required to maintain the Offboard mode flight state between Jetson-nano and Pixhawk.

Public_node receives the current operating value of the aircraft via topic '/SetpointPose_UAV' and continues to maintain the data communication flow between the embedded system and Pixhawk via topic '/setpoint_position' in MAVROS. The communication cycle was set at 10 Hz to ensure the flight stability of the Offboard mode with minimal load through multiple checks.

Verifty_node is designed to ensure the safety of design tests, monitoring the PWM pulse width of radio channel 7 via MAVROS topic '/rc/in'. When the channel is activated, the pulse width is changed and the airplane is controlled by the UAV's flight mode service. This function allows flight mode control to be switched manually or vice versa on the Offboard, allowing control of the aircraft by intervening in unexpected situations during autonomous missions.

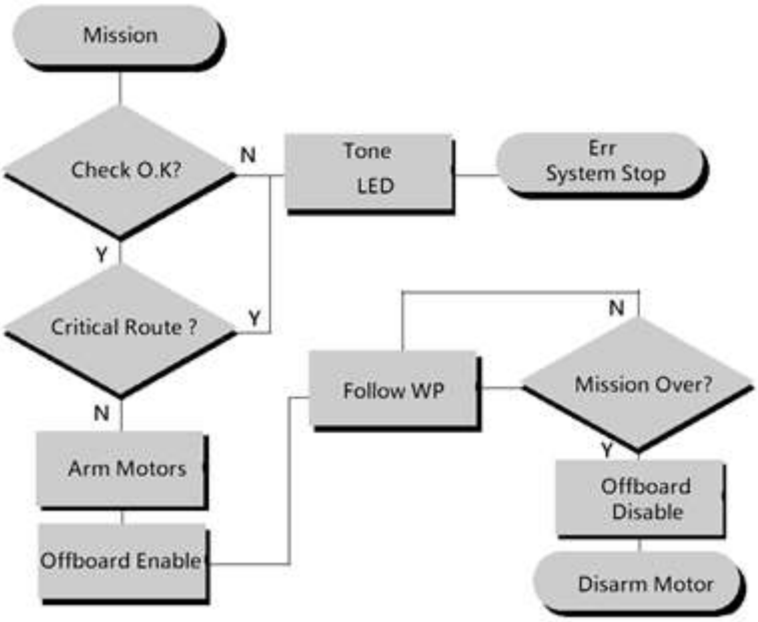

Fig. 4: Functional Flowchart

4 Implementation of a self-contained, embedded system

4.1 SITL/HITL Simulation

For control codes written for autonomous flight and collision avoidance, the environment in Figure 5 was constructed and pre-simulated using the function of ROS to safely identify problems and inconsistencies in the aircraft's Pixhawk Firmware and embedded systems.

Software in the loop (SITL) simulations were performed to verify the integrity of the algorithm, and validation was performed through Hardware in the loop (HITL) to evaluate actual hardware control, integration between Pixhawk and Jetson nano.

Fig. 5: HITL Simulation Environment

In the simulation, the aircraft is replaced by the mathematical model provided by the ROS simulator, but simulation control is provided to check all functions with respect to the control algorithm.

The simulation was carried out in two different ways. The first is the Mission Planner in Figure 6, which specifies a square flight path of 100 m away from the take-off point, maintaining an altitude of 15 m, and analyzing the results using Gazebo Logging data and ROS rqt topic monitoring data.

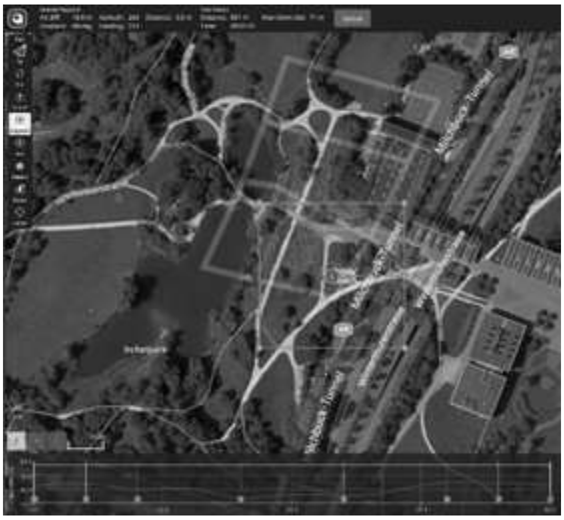

Fig. 6: Mission Planning (QGC)

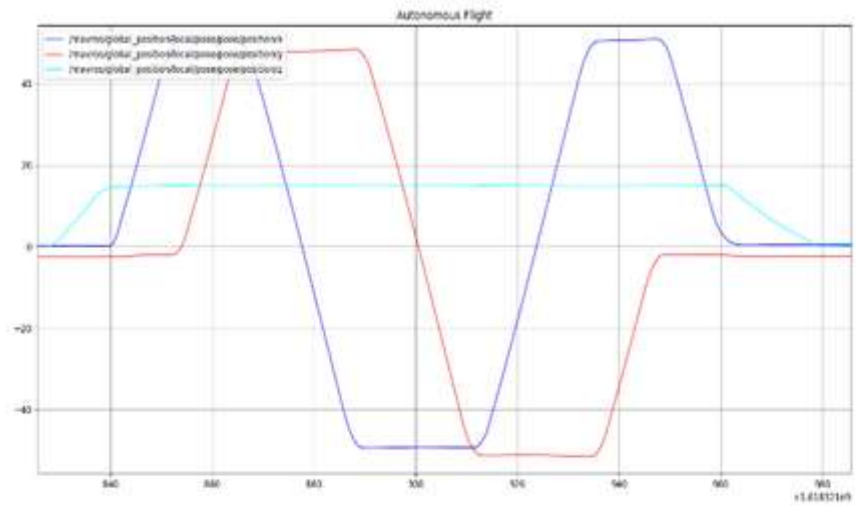

Fig. 7: HITL Simulation Results

The flight was carried out at a speed of 2 m/s and the effects of the wind were ignored, confirming that the aircraft flew exactly following the path points set in the mission plan.

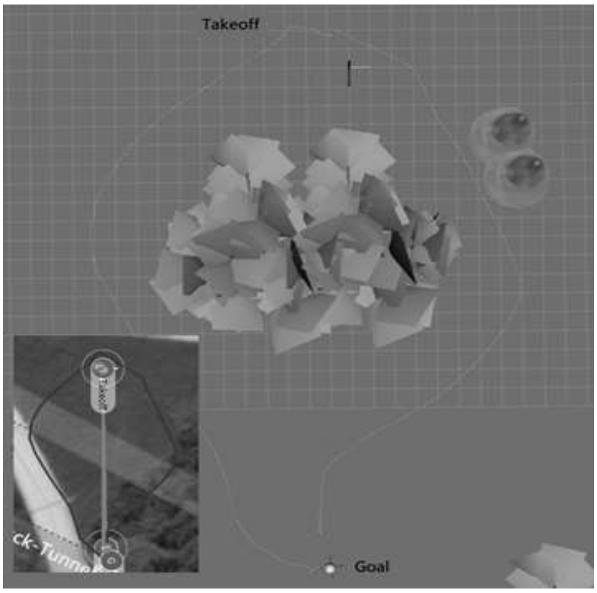

Fig. 8: Collision Avoidance Example

Secondly, the flight path follow-up status for the application and optimization of the avoidance algorithm under development using the SITL simulation open source of ROS containing obstacles was confirmed as shown in Figure 8.

5 CONCLUSION AND FUTURE RESEARCH PLANS

MAV is drawing attention because it is the smallest remotely controlled drone and is easy to use for various environments and purposes. In this study, we implement a mounted embedded system that enables autonomous mission performance by minimizing operator intervention even in obstacle environments to meet this purpose. Based on similar research cases, simulations were conducted using ROS support packages to apply obstacle detection/avoidance algorithms and to increase safety and efficiency in the research and development process of the system, and the performance of obstacle detection/avoidance and route tracking was verified.

It is predicted that it will be possible to implement an embedded system optimized for performing MAV's safe self-flight mission through additional research such as a three-dimensional obstacle avoidance path determination algorithm considering the application of multiple sensors and real-time system processing.

REFERENCES

[1] Kim Kyoung-Ho et al. “3D Library Platform Construction using Drone Images and its Application to Kangwha Dolmen” Cartoon and Animation Studies, Ser No.48, KOSCAS, pp. 199-215, 2017.

[2] Nagarjuna, K., Suresh, G.R. “Design of effective landing mechanism for fully autonomous Unmanned Aerial Vehicle”, International Conference on Signal Processing, Communication and Networking (lCSCN), pp 1-6, 2015.

[3] M. A. Ma'sum et al. "Simulation of intelligent Unmanned Aerial Vehicle (UAV) For military surveillance," 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 161-166, 2013.

[4] Amazon.com: Amazon Prime Air, 2016. http://www.amazon.com/b?node=8037720011

[5] Kie-Jeong Seong et al. “Development Trend of the Autonomous Flight Control Technology”, 2008 Aerospace Industry Technology Trend Vol. 62 pp. 143~153.

[6] The Open Source Robotics Foundation: Robot Operating System (ROS), http://www.ros.org/

[7] Engel, J., Sturm, J., Cremers, D.: Camera-based navigation of a low-cost quadrocopter. In: International Conference on Intelligent Robots and Systems (IROS), pp. 2815–2821, 2012.

[8] Meier, L., Honegger, D., Pollefeys, M.: PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: IEEE International Conference on Robotics and Automation, pp. 6235–6240, 2015.

[9] Cocchioni, F., Pierfelice, V., Benini, A., Mancini, A., Frontoni, E., Zingaretti, P., Ippoliti, G., Longhi, S.: Unmanned ground and aerial vehicles in extended range indoor and outdoor missions. International Conference on Unmanned Aircraft Systems, pp. 374–382, 2014.

[10] Johann Borenstein, “Real-Time Obstacle Avoidance for Fast Mobile Robots”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. 19, NO. 5, SEPTEMBER/OCTOBER, 1989.

[11] Simon Vanneste, Ben Bellekens, Maarten Weyn, “3DVFH+: Real-Time Three-Dimensional Obstacle Avoidance Using an Octomap”, CEUR Workshop Proceedings. 2014.