Autonomous Flight of a Quadrocopter Group with the Use of the Virtual Leader Strategy

Philipp Samoilov
North Caucasus
Federal University,
Department of Information
Systems & Technologies,
Stavropol, Russia
fsamoilov@ncfu.ru

Dmitry Vinokursky
North Caucasus
Federal University,
Department of Information
Systems & Technologies,
Stavropol, Russia;
dlvinokursky@gmail.com

Marco Schaerf
Sapienza Universita
di Roma,
Rome, Italy
marco.schaerf@uniroma1.it

Massimo Mecella
Sapienza Universita
di Roma,
Rome, Italy
mecella@dis.uniroma1.it

Abstract

In this article we present an algorithm of controlling the quadrocopters swarm and a theory of applying the Kalman filter for the equations of motion of a quadrocopter in mountainous conditions. In our case, in order to coordinate the group, it is necessary to form the spatial programmatic trajectory of the UAV using the appropriate control law. The concept of coordinated reversal is introduced, which allows to obtain analytical equations of spatial motions expressed through the definition of the velocity vector and the yaw angle. The algorithm was tested in the Gazebo simulator. The results are used for spatial motion of quadrocopter groups.

1 Introduction

Currently, there is an increasing interest in the use of UAVs in a number of practical applications. Nowadays, many inexpensive and easy-to-assemble UAV models with powerful on-board computers and auxiliary portable computers on the board are available on the market.

One of the most popular types used in the commercial field and for academic research is the quadrocopter. Quadrocopters equipped with onboard sensors are popular because of their size, the ability to hover over objects, to navigate in 3D space. They can be used in various fields ranging from monitoring [1], [2], [4] to searching and implementing rescue operations [3]. The use of these UAVs in mountain areas has the following advantage:

In: S. Hölldobler, A. Malikov (eds.): Proceedings of the YSIP-3 Workshop, Stavropol and Arkhyz, Russian Federation, 17-09-2019–20-09-2019, published at http://ceur-ws.org

it can be located where human presence is impossible. At the same time, in mountain conditions, it may be difficult to receive signals from the GPS / GLONASS navigation systems, as well as distortion of signals from other sensors. In the presented work, the Kalman filter is used to reduce the errors of signals in the UAV control system in mountain conditions. The combined quadrocopter groups or swarms that can cover large space, collect more information and are less susceptible to the loss of a group member. Groups control can be carried out using different strategies, each of them has advantages and disadvantages. There are several examples of successful use of swarm strategies for controlling quadrocopter groups. Kushleyev developed a swarm of 20 miniature quadrocopters capable of forming several formations using centralized strategies [5]. Burkle implemented a swarm of quadrocopters capable of flying in external spaces, also using the centralized method and a ground station as a control device [6]. Vsrhelyi used swarm rules for the fly of 10 quadrocopters in external environments [7]. In [8], a group of simulation UAVs was used for monitoring. The most used algorithms found in the literature can be classified into three types: leader-follower [9], [10], virtual structure [11], [12] and behavioral approach [13], [14]. In the first type, one of the swarm is selected as the leader of the group, the other participants change their position in the group according to the position of the leader. This approach has an advantage, as it simplifies the task for the operator who controls only the leader, however, in the cases of a leader failure, the control of the entire group will be lost. A virtual structure is a consideration of the whole group as one fixed structure which is represented as the one control point. The behavioral approach represents a programmed motion of the group in a certain way, for example, avoiding collisions, or moving as close as possible to each other. In most cases, this type of strategy is based on real natural phenomena [15]. In order to control the swarm and maintain the formation, we suggest using an approach based on a virtual leader.

2 The equation of the quadrocopter motion

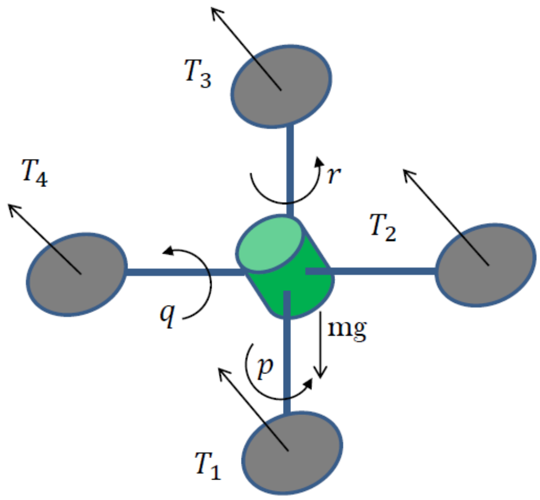


Figure 1: Simplied quadrotor model

The fixed coordinate system XYZ is given. The dynamics of the quadrocopter motion will be considered as a set of the translational motion of the center of the quadrocopter mass and the rotational motion of the quadrocopter relatively to the center of mass. This motion is described by the following equations:

$$\begin{cases} m\begin{pmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{pmatrix} = \begin{pmatrix}0\\0\\-mg\end{pmatrix} + R\begin{pmatrix}0\\0\\ \sum F_i\end{pmatrix} \\ I\begin{pmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{pmatrix} = \begin{pmatrix} l(F_2-F_4)\\ l(F_3-F_1)\\ -Q_1+Q_2-Q_3+Q_4\end{pmatrix} \end{cases} - \begin{pmatrix}p\\q\\r\end{pmatrix} \times I \begin{pmatrix}p\\q\\r\end{pmatrix} \quad (1)$$

Here

$$R = \begin{pmatrix} \cos\psi\cos\theta - \sin\varphi\sin\psi\sin\theta & -\cos\varphi\sin\psi & \cos\psi\sin\theta + \cos\theta\sin\psi\sin\varphi \\ \cos\theta\sin\psi + \cos\psi\sin\varphi\sin\theta & \cos\varphi\cos\psi & \sin\varphi\sin\theta - \cos\psi\cos\theta\sin\varphi \\ -\cos\varphi\sin\theta & \sin\varphi & \cos\varphi\cos\theta \end{pmatrix} \tag{2}$$

R - the matrix of rotation, φ, ψ, θ - Euler angles, T_i - thrust, Q_l – torgue

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & -\cos\varphi\sin\theta \\ 0 & 1 & \sin\varphi \\ \sin\theta & 0 & \cos\varphi\cos\theta \end{pmatrix} \begin{pmatrix} \dot\varphi \\ \dot\theta \\ \dot\psi \end{pmatrix} \tag{3}$$

m – the mass of quadrocopter

$$I = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \tag{4}$$

The matrix of inertia moments.

The solution of system (1) completely determines the quadrocopter motion trajectory.

In the conditions of mountain areas there is a high probability of noise in the operation of sensors (strong wind, weak electromagnetic signal, etc) in this case, it is desirable to correct the transfer of coordinates to the control system with the help of the filter. This can be an alpha-beta filter or a Kalman filter.

Consider the work of the Kalman filter in detail.

In general, the Kalman filter system for a discrete system is:

$$x_k = \Phi_{k-1} x_{k-1} + w_k,$$

$$z_k = {}_k x_k + v_k.$$

Here

x_k – the dynamic system state vector,

z_k – the measurements at the time,

t_k, w_k – the noise of the system,

v_k – the noise of the measurements.

The task of the Kalman filter is to determine the estimate of the state vector x_k, this estimate we denote $\hat{x}_k$, which is a function of the measurement $z_1, ..., z_k$ and minimizes mean-root-square error.

The Kalman filter algorithm works according to the predictor corrector scheme (prediction correction). The algorithm of the Kalman filter can be presented in more detail. In the moment of time t_{k-1} there is an the estimate of the state vector $\hat{x}_{k-1}$ and we need to receive a vector x_k in the moment of time t_k.

A predictive estimate of $\hat{x}_k^-$- is built, based on the estimate of $\hat{x}_{k-1}^+$ of the previous step. We obtain the measurement z_k and the correction of the estimate occurs at the moment of time t_k. As a result, we have an estimate of the state vector at some step $\hat{x}_k^+$ (fig.2). So "-" means a priori estimate. The sign "+" posterior estimate.

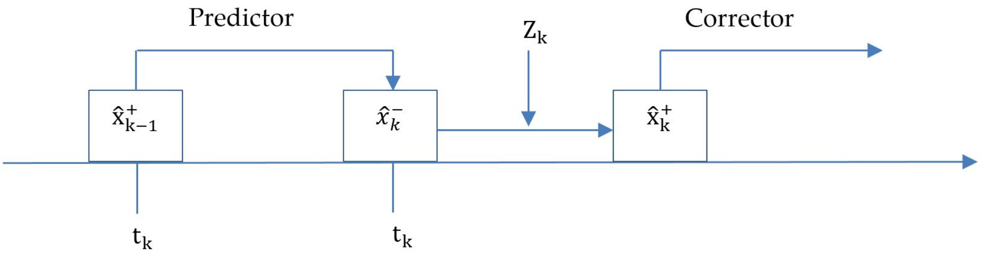


Figure 2: The scheme of Kalman filter working

3 Hardware

Our system consists of 2 UAVs and a ground station communicating with the group via a Wi-Fi protocol. The ground station is responsible for collecting information from each UAV and sends information about the desired position of the quadrocopter in the space. The goal is to send a group of quadrocopters to the desired point avoiding collisions and keeping the group structure. The quadrocopter has a simple design with 4 engines, two of them rotate clockwise, two anticlockwise.

It is capable of taking off and landing (VTOL) and having 6 degrees of freedom. The motion of the duadrocopter in space is carried out by adjusting the thrust.

By adjusting the speed of engines rotation, we can move the quadrocopter relatively to the yaw angle. The qudrocopter is moved depending on the pitch and roll angles by adjusting the engine moments. The assembled quadrocopters designed for the testing of group control algorithms have a 650 mm frame with 490 Kv engines, each of them is connected to its own electronic speed controller (ESC) which is connected to the flight controller via the I2C bus. Each UAV has an external GPS which is very important for applying our algorithm. The used flight controller is Pixhawk.

3.1 Software architecture

The Pixhawk is an open source architecture [16] equipped with a variety of sensors such as a gyroscope, an accelerometer, a magnetometer and a barometer, with the ability to connect an external GPS module. The software which is running on the flight controller is called the PX4. This software allows you to interpret information received from sensors and signals from a manual control station, generates flight data and monitors each engine. The main advantage of the Pixhawk is to provide low-level stabilization and control of UAVs. A useful feature of the Pixhawk is the ability to interact with other devices using the MAVLink protocol developed specifically for UAV applications. We can also use this protocol to get autonomous control of a UAV using a companion computer, such as the Raspberry Pi for communication with the flight controller. In addition, you can use it to read flight information and send various commands to control the flight controller.

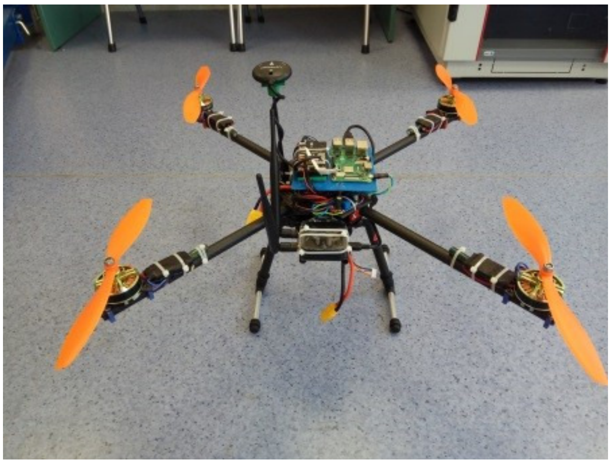


Figure 3: The quadrocopter used in laboratory

Typically, the Pixhawk is controlled by a manual control station through a radio channel. However, in our case, the main control is carried out via a companion computer (Raspberry Pi with the installed OS Raspbian) which is mounted on the board and connected to the flight controller via the UART interface. Communication with the ground station is carried out through an external Wi-Fi module which is connected to the companion computer. Thus, we can use the onboard computer as a tool for autonomous flights, with the ability to switch to a manual control station in case of emergency.

3.2 Details of the mission

We define the global system as a Cartesian coordinate system fixed on the ground. Each UAV uses its sensors to determine its position within the global coordinate system. Considering the geometry of the earth and the originality of the global frame, the latitude and longitude obtained from the GPS sensor can be converted into a value in meters into a position along the X and Y axes. This will allow us to determine the initial position of the UAV. Next, the flight controller uses the Extended Kalman filter to estimate the position which is obtained from the flight controllers sensors. The Z coordinate is calculated from the total measurements received from the GPS and barometer. Each UAV gets a unique ID. The UAV group is viewed as a set of positions relative to the global frame. The distribution of positions in the group is carried out according to ID.

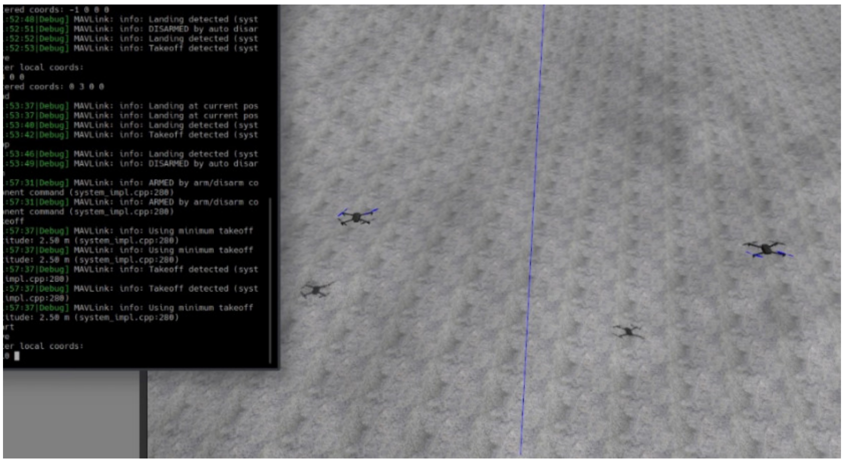
4 Software

The software is represented as a console utility running on OS Arch Linux. The software was developed in C++17. The software allows controlling the UAV using a ground-to-air system: UAVs are connected in a star topology, where the ground station is represented as a central point with the running software. UAVs send telemetry data to the ground station, where, the position of UAV is calculated in the air by sending data back to the UAV. The main control is carried out by setting the velocity vector of the UAV, as well as the angle of yaw rotation. The ground station monitors the change and adjusts the velocity vector to achieve optimal performance of the task. At this stage, the task is supposed to set the target flight coordinates of the UAV group in the projected coordinate system: the group position is changed by specifying a vector of three coordinates Δ x, Δy, Δz, where the positive increments are directed towards the east, north and upwards respectively. In addition to the linear displacement vector, it is possible to set the angle of rotation of the swarm in degrees as a true azimuth. In order to abstract and simplify the development of the communication module with UAVs, the DroneCode SDK library has been applied. With the help of this library, it is possible to implement a fast connection to UAVs using the MAVLink v2 protocol, using both TCP-UDP protocols and serial ports. Among other things, it is possible to track errors that occur in the process of conducting experiments. The software allows to create various methods of swarm manipulation. These methods include various ways of forming and moving the group. In order to debug a forming group UAVs the simple methods are used. The group is located in the same horizontal plane and kept at equal distances from the virtual center of the swarm. At the top level of abstraction, the control of this center is given, after which the necessary calculations are made for each of the agents. In order to move the group the velocity vector setting of the UAV is performed.

The process of group monitoring is controlled by the ground station. The accuracy of motion of each agent is checked and its deviation in a given radius is corrected. At the end of the motion, the UAVs correct their positions in the group according to the heterogeneity of the used devices: the difference in mass, dimensions and force of the thrust.

5 Simulation

The algorithm was tested in a software simulator. All simulations were performed in the Gazebo simulator, which allows to simulate the 3D spaces, robots and sensors. The main advantage of Gazebo is the ability to integrate various platforms. Using the Pixhawk and Gazebo software, we have the opportunity to create a simulation close to reality and test our algorithm as on a real device.



```
ered coords: -1 0 0 0
:52:48|Debug] MAVLink: info: Landing detected (syst
:52:51|Debug] MAVLink: info: DISARMED by auto disar
:52:52|Debug] MAVLink: info: Landing detected (syst
:52:53|Debug] MAVLink: info: Takeoff detected (syst
e
er local coords:
 0 0
ered coords: 0 3 0 0
d
:53:37|Debug] MAVLink: info: Landing at current pos
:53:37|Debug] MAVLink: info: Landing at current pos
:53:40|Debug] MAVLink: info: Landing detected (syst
:53:42|Debug] MAVLink: info: Takeoff detected (syst
p
:53:46|Debug] MAVLink: info: Landing detected (syst
:53:49|Debug] MAVLink: info: DISARMED by auto disar

:57:31|Debug] MAVLink: info: ARMED by arm/disarm co
nent command (system_impl.cpp:280)
:57:31|Debug] MAVLink: info: ARMED by arm/disarm co
nent command (system_impl.cpp:280)
eoff
:57:37|Debug] MAVLink: info: Using minimum takeoff
itude: 2.50 m (system_impl.cpp:280)
:57:37|Debug] MAVLink: info: Using minimum takeoff
itude: 2.50 m (system_impl.cpp:280)
:57:37|Debug] MAVLink: info: Takeoff detected (syst
impl.cpp:280)
:57:37|Debug] MAVLink: info: Takeoff detected (syst
impl.cpp:280)
:57:37|Debug] MAVLink: info: Using minimum takeoff
itude: 2.50 m (system_impl.cpp:280)
rt
e
er local coords:
0 █
```

Figure 4: The simulation results

6 6 Conclusions

This article presents the software that allows to control a group of quadrocopters based on Pixhawk flight controllers. The software was developed in the C++ environment and runs on the Arch Linux OS. In a simulation environment based on a Gazebo simulator, the algorithm was launched and tested under conditions as close to reality as possible. Five simulation flights were performed. Each of the examples had different starting GPS positions. We defined a scenario where two quadrocopters with different identification numbers flew in a line at a height of 2.5 meters. The application of this algorithm is quite simple, and some improvements can be made in the future. In the software developed by us, such functions are implemented that allow the group to fly in a straight line, without the possibility of a system reversal, and further, with the realization of this possibility, the group's maneuverability will increase significantly. Another interesting direction will be the implementation of the function to identify fixed obstacles and avoiding collisions with them.

References

[1] Merino, L., Caballero, F., De Dios, J. R. M., Maza, I., Ollero, A. *Automatic forest fire monitoring and measurement using unmanned aerial vehicles, Proceedings of the 6th International Congress on Forest Fire Research*: 2010.

[2] Ollero, A., Merino, L. *Unmanned aerial vehicles as tools for forest-fire fighting, Forest Ecology and Management*: p. 263, 2006.

[3] Baker, C., Ramchurn, G., Teacy, L., Jennings, N. *Planning search and rescue missions for UAV teams*: 2016.

[4] Geng, L., Zhang, Y. F., Wang, P. F., Wang, J. J., Fuh, J. Y., Teo, S. H. *UAV surveillance mission planning with gimbaled sensors, 11th IEEE International Conference on Control & Automation (ICCA)*: 320-325, 2014.

[5] Kushleyev, A., Mellinger, D., Powers, C., Kumar, V. *Towards a swarm of agile micro quadrotors, Autonomous Robots*: 287-300, 2013.

[6] Burkle, A., Segor, F., Kollmann, M. *Towards autonomous micro uav swarms, Journal of intelligent & robotic systems*: 339-53, 2001.

[7] Vas arhelyi, G., Vir agh, C., Somorjai, G., Tarcai, N., Sz or enyi, T., Nepusz, T., Vicsek, T. *Outdoor flocking and formation flight with autonomous aerial robots, IEEE/RSJ International Conference on Intelligent Robots and Systems*: 3866-3873, 2014.

[8] De Benedetti, M., DFLUrso, F., Messina, F., Pappalardo, G., Santoro, C. *UAV-based Aerial Monitoring: A Performance Evaluation of a SelfOrganising Flocking Algorithm, 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*: 248-255, 2015.

[9] Gu, Y., Seanor, B., Campa, G., Napolitano, M. R., Rowe, L., Gururajan, S., Wan, S. *Design and flight testing evaluation of formation control laws, IEEE Transactions on Control Systems Technology*: 1105-1112, 2006.

[10] Ambroziak, L., Gosiewski, Z. *Two stage switching control for autonomous formation flight of unmanned aerial vehicles—*, Aerospace Science and Technology 46: 221-226, 2015.

[11] Egerstedt, M., Hu, X., Stotsky, A. *Control of mobile platforms using a virtual vehicle approachm, IEEE transactions on automatic control*: 1777-1782, 2001.

[12] Askari, A., Mortazavi, M., Talebi, H. A. *UAV formation control via the virtual structure approach.* Journal of Aerospace Engineering: v. 28, n. 1, p. 04014047, 2013.

[13] Balch, T., Arkin, R. C. *Behavior-based formation control for multirobot teams, IEEE transactions on robotics and automation*: 926-939, 1998.

[14] Virgh, C., Vsrhelyi, G., Tarcai, N., Szrnyi, T., Somorjai, G., Nepusz, T., Vicsek, T. *Flocking algorithm for autonomous flying robots.* Bioinspiration & biomimetics: v. 9, n. 2, p. 025012, 2014.

[15] Braga, R., Ramos, A. C. B., da Silva, R. C., Mora-Camino, F. *Collision avoidance based on Reynolds Rules: a case study using quadrotors*. 14th International Conference on Information Technology : New Generations: 2017.

[16] Pixhawk homepage: https://pixhawk.org/. Accessed: September, 2016.