Simulation and implementation of alternative attitude control algorithms for a micro multirotor flying platform

Abstract

The most popular control algorithm in attitude stabilization of multirotor aerial vehicles is a PID controller. It is used so frequently because of low computational requirements and simplicity of implementation and tuning. However, there are many other control algorithms suitable for this task, for example fuzzy logic or neural networks controllers. These algorithms are more complex than PID, but with appropriate knowledge and hardware it is possible to implement them in the on-board controller. This paper presents the comparison of the mentioned algorithms during simulation experiments. The construction process of a micro multirotor flying platform NeuroQuad and the implementation of the tested algorithms on this platform are described.

Keywords: quadrotor, control, fuzzy logic, ANN, PID, NeuroQuad.

1. Introduction

Evaluation of quadrotor control algorithms is recently a very popular issue [1], [2]. The most important part of this kind of software is attitude stabilization, which is vital for flight realization.

In this paper, the comparison of different attitude control algorithms is presented. For this purpose three types of control methods were considered. Firstly, the standard PID controller approach, precisely a PD structure. Motivation for this solution was a simplicity of tuning and control, but also reference acquisition for comparison to the next algorithms performance. Secondly, fuzzy logic and finally the neural network controller were designed. Robustness and ability to approximate these algorithms are their advantages. PD and fuzzy controllers were tuned empirically, based on the knowledge acquired from previous projects. Data from simulating these two controllers was used to train the artificial neural network controller.

This project was realized under the masters degree thesis. There were two main goals of the thesis, to test the mentioned control algorithms in simulation model and implement them in the developed platform.

The paper is organized as follows. The description of the mathematical and simulation model is presented in the next section. The performed simulation experiments are shown in the third section. The fourth section contains the introduction to the platform development process and description of the implementation of the developed control algorithms. The paper is summarized in the last section.

2. Simulation model

The first step of developing a simulator is to acquire and calibrate the mathematical model of a quadrotor robot. One of the most complete models is included in Robotics Toolbox from P. Corke [5] for MATLAB. This model was developed by P. Pounds in his doctoral thesis [4]. It covers many physic and aerodynamic forces, for example the flapping of rotors and a drag torque. After the mathematical model specification, the Simulink simulation development was undertaken (Fig. 1). It consists of four different sections, the most important ones are described in the following sections. A simulation period was set to 2.5 ms, the same as control the loop refresh rate in the developed microavionics on the robot described in Section 4. Calibration process of the mathematical model consists of measuring specific physical parameters of the developed platform and implementing them into the simulation model, starting from the mass of the robot through the thrust coefficient and moments of inertia, ending with the estimated hub clamp mass.

It is worth mentioning that the mathematical model coordinate frame is located alongside the robot arms, but a more intuitive version for the pilot is a frame location between those arms. Hence, the rotation of the robot coordinate system by 45° was performed.

Also conversion from radians to degrees is realized before the regulator block.

The vector of preset values is divided into four inputs: Roll, Pitch, Yaw and height above the ground. The first three are related to the rotated coordinated system and expressed in degrees, the height is expressed in meters. The authors assumed that Pitch angle had the same preset signal as Roll, but with a 10 seconds delay. This allowed testing attitude stabilization in both axes.

The paper is organized as follows. The description of the mathematical and simulation model is presented in the next section. The performed simulation experiments are shown in the third section. The fourth section contains the introduction to the platform development process and description of the implementation of the developed control algorithms. The paper is summarized in the last section.

2. Simulation model

The first step of developing a simulator is to acquire and calibrate the mathematical model of a quadrotor robot. One of the most complete models is included in Robotics Toolbox from P. Corke [5] for MATLAB. This model was developed by P. Pounds in his doctoral thesis [4]. It covers many physic and aerodynamic forces, for example the flapping of rotors and a drag torque. After the mathematical model specification, the Simulink simulation development was undertaken (Fig. 1). It consists of four different sections, the most important ones are described in the following sections. A simulation period was set to 2.5 ms, the same as control the loop refresh rate in the developed microavionics on the robot described in Section 4. Calibration process of the mathematical model consists of measuring specific physical parameters of the developed platform and implementing them into the simulation model, starting from the mass of the robot through the thrust coefficient and moments of inertia, ending with the estimated hub clamp mass.

It is worth mentioning that the mathematical model coordinate frame is located alongside the robot arms, but a more intuitive version for the pilot is a frame location between those arms. Hence, the rotation of the robot coordinate system by 45° was performed. Also conversion from radians to degrees is realized before the regulator block.

The vector of preset values is divided into four inputs: Roll, Pitch, Yaw and height above the ground. The first three are related to the rotated coordinated system and expressed in degrees, the height is expressed in meters. The authors assumed that Pitch angle had the same preset signal as Roll, but with a 10 seconds delay. This allowed testing attitude stabilization in both axes.

The paper is organized as follows. The description of the mathematical and simulation model is presented in the next section. The performed simulation experiments are shown in the third section. The fourth section contains the introduction to the platform development process and description of the implementation of the developed control algorithms. The paper is summarized in the last section.

2. Simulation model

The first step of developing a simulator is to acquire and calibrate the mathematical model of a quadrotor robot. One of the most complete models is included in Robotics Toolbox from P. Corke [5] for MATLAB. This model was developed by P. Pounds in his doctoral thesis [4]. It covers many physic and aerodynamic forces, for example the flapping of rotors and a drag torque. After the mathematical model specification, the Simulink simulation development was undertaken (Fig. 1). It consists of four different sections, the most important ones are described in the following sections. A simulation period was set to 2.5 ms, the same as control the loop refresh rate in the developed microavionics on the robot described in Section 4. Calibration process of the mathematical model consists of measuring specific physical parameters of the developed platform and implementing them into the simulation model, starting from the mass of the robot through the thrust coefficient and moments of inertia, ending with the estimated hub clamp mass.

It is worth mentioning that the mathematical model coordinate frame is located alongside the robot arms, but a more intuitive version for the pilot is a frame location between those arms. Hence, the rotation of the robot coordinate system by 45° was performed. Also conversion from radians to degrees is realized before the regulator block.

The vector of preset values is divided into four inputs: Roll, Pitch, Yaw and height above the ground. The first three are related to the rotated coordinated system and expressed in degrees, the height is expressed in meters. The authors assumed that Pitch angle had the same preset signal as Roll, but with a 10 seconds delay. This allowed testing attitude stabilization in both axes.

The paper is organized as follows. The description of the mathematical and simulation model is presented in the next section. The performed simulation experiments are shown in the third section. The fourth section contains the introduction to the platform development process and description of the implementation of the developed control algorithms. The paper is summarized in the last section.

2. Simulation model

The first step of developing a simulator is to acquire and calibrate the mathematical model of a quadrotor robot. One of the most complete models is included in Robotics Toolbox from P. Corke [5] for MATLAB. This model was developed by P. Pounds in his doctoral thesis [4]. It covers many physic and aerodynamic forces, for example the flapping of rotors and a drag torque. After the mathematical model specification, the Simulink simulation development was undertaken (Fig. 1). It consists of four different sections, the most important ones are described in the following sections. A simulation period was set to 2.5 ms, the same as control the loop refresh rate in the developed microavionics on the robot described in Section 4. Calibration process of the mathematical model consists of measuring specific physical parameters of the developed platform and implementing them into the simulation model, starting from the mass of the robot through the thrust coefficient and moments of inertia, ending with the estimated hub clamp mass.

It is worth mentioning that the mathematical model coordinate frame is located alongside the robot arms, but a more intuitive version for the pilot is a frame location between those arms. Hence, the rotation of the robot coordinate system by 45° was performed. Also conversion from radians to degrees is realized before the regulator block.

The vector of preset values is divided into four inputs: Roll, Pitch, Yaw and height above the ground. The first three are related to the rotated coordinated system and expressed in degrees, the height is expressed in meters. The authors assumed that Pitch angle had the same preset signal as Roll, but with a 10 seconds delay. This allowed testing attitude stabilization in both axes.

The paper is organized as follows. The description of the mathematical and simulation model is presented in the next section. The performed simulation experiments are shown in the third section. The fourth section contains the introduction to the platform development process and description of the implementation of the developed control algorithms. The paper is summarized in the last section.

2. Simulation model

The first step of developing a simulator is to acquire and calibrate the mathematical model of a quadrotor robot. One of the most complete models is included in Robotics Toolbox from P. Corke [5] for MATLAB. This model was developed by P. Pounds in his doctoral thesis [4]. It covers many physic and aerodynamic forces, for example the flapping of rotors and a drag torque. After the mathematical model specification, the Simulink simulation development was undertaken (Fig. 1). It consists of four different sections, the most important ones are described in the following sections. A simulation period was set to 2.5 ms, the same as control the loop refresh rate in the developed microavionics on the robot described in Section 4. Calibration process of the mathematical model consists of measuring specific physical parameters of the developed platform and implementing them into the simulation model, starting from the mass of the robot through the thrust coefficient and moments of inertia, ending with the estimated hub clamp mass.

It is worth mentioning that the mathematical model coordinate frame is located alongside the robot arms, but a more intuitive version for the pilot is a frame location between those arms. Hence, the rotation of the robot coordinate system by 45° was performed. Also conversion from radians to degrees is realized before the regulator block.

The vector of preset values is divided into four inputs: Roll, Pitch, Yaw and height above the ground. The first three are related to the rotated coordinated system and expressed in degrees, the height is expressed in meters. The authors assumed that Pitch angle had the same preset signal as Roll, but with a 10 seconds delay. This allowed testing attitude stabilization in both axes.

The paper is organized as follows. The description of the mathematical and simulation model is presented in the next section. The performed simulation experiments are shown in the third section. The fourth section contains the introduction to the platform development process and description of the implementation of the developed control algorithms. The paper is summarized in the last section.
2.1. Regulator and mixer

These two blocks represent control algorithms which later on will be implemented in a microavionics system. The Regulator block consists of four separate controllers, one for each angle (Roll, Pitch and Yaw) and one for altitude control. Yaw and altitude controllers were not the main target of the research, therefore the tuning of these controllers was performed only at the beginning of the experiments and their parameters remained unchanged. For the yaw angle PD controller there was used a PID version for altitude, because of the need to eliminate a steady state error.

The remaining two regulators (roll and pitch) were changing depending on the experimental stage. The first two cases, PD and fuzzy logic, were tuned empirically, finally an ANN (Artificial Neural Network) regulator was taught based on the data from simulations of the previous ones.

Each controller has the independent output from the Regulator block. These values are combined by the function Mixer, which distributes the power among four motors in the range of 0÷100%.

2.2. BLDC and quadrotor

This section describes simulation of the physical behavior of the quadrotor. The BLDC block transforms signals from Mixer to the rotation frequency of every propulsion unit, simulating the behavior of the BLDC (Brushless Direct Current) motors. This dependency was measured on a constructed testing bench for these propulsion units. During tests on this equipment the crucial data: voltage, current, thrust, rotation frequency and PWM duty cycle, were measured. This was one of the required experiments to calibrate the mathematical model. Unfortunately, the authors were not able to gather dynamical characteristics and because of that were not the main target of the research, therefore the tuning of those variations were chosen empirically.

As mentioned before, the Quadrotor block is a mathematical model of the quadrotor robot. It takes four rotation frequencies as the input and gives the position and attitude with all derivatives as the output.

3. Simulation model

The main goal of this research was to compare performance of different control algorithms. Three types of controllers were implemented: PD, fuzzy logic and ANN. As mentioned earlier, the PD and the fuzzy logic controller were tuned empirically. All the regulators received the error and its derivative as an input. The error was calculated by subtracting a recent angle from the preset value. A classical structure of the PD controller was used; after multiplication both elements are added as an output. In the fuzzy logic regulator, the mamdani inference method was used with five membership functions in each input and with centroid defuzzification in the output. The inputs to the controller were scaled to the range –1÷1 and the output to –40÷40.

Four adequately created control sequences were used to test the tuned algorithms. Those signals included low and high as well as slow and fast changes of the preset values, which allowed checking the robustness of the presented controllers.

During every test sequence, input and output signals from the PD and fuzzy logic controllers were gathered. Those data later on were used in the learning and validating process of the ANN controller, but the first step was to select the number of hidden neurons. After a series of testing, every case from one to ten hidden neurons was evaluated by IAE (Integral Absolute Error) and ISE (Integral Square Error) control measures during learning as well as validating sequences. Finally, the structure of a neural network with the best performance was selected, two input neurons, two in a hidden layer and one output neuron.

After the mentioned steps, all the control algorithms were tuned and could be compared mutually. The final results of comparison are shown in Tab. 1.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>IAE</th>
<th>ISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td>0.4841</td>
<td>0.08264</td>
</tr>
<tr>
<td>Fuzzy (Centroid)</td>
<td>0.1472</td>
<td>0.02292</td>
</tr>
<tr>
<td>ANN</td>
<td>0.1549</td>
<td>0.02519</td>
</tr>
</tbody>
</table>

The best results were achieved for the fuzzy logic controller, those for the ANN regulator were nearly the same. Unfortunately, the PD controller was three times worse. Moreover, such a result of the fuzzy logic controller was sacrificed in terms of the execution time. It was almost twice as long as that of the other algorithms. The artificial neural network was the fastest one. To sum up, the neural network controller had the best performance to execution time ratio.

4. Platform development

The second main goal of the thesis was to construct a physical platform and to implement the simulated algorithms. Firstly, the whole robot was created using CAD software. Thanks to this approach many parameters needed for a mathematical model, such as moments of inertia and the center of gravity, could be calculated.

4.1. Hardware

The platform frame is based on classical cross shaped design and it is made of CFRP (Carbon Fiber Reinforced Polymer). A 13 mm diameter BLDC motor is mounted with a 3 inch propeller at the end of each arm. Wires from motors are soldered to ESCs (Electronic Speed Controllers) which are connected to microavionics. This electronic module is screwed to the robot frame and a battery is attached underneath it. Under each motor, 3D printed supporting legs are mounted. The ready to flight platform is presented in Fig. 2.

4.2. Electronics

The microavionics controller and a radio dongle were designed by the authors using PCB designing software. Two nRF24L01 radio modules are used to communicate between the robot and the dongle which is connected to a PC. The pilot uses a game pad connected to the PC, next the program passes orders to the radio dongle which sends them to the robot.

The avionics and the motors are powered by a single cell lithium-polymer battery. An inertial sensor and a barometer were used to calculate the attitude and altitude estimates [6, 7].
A microcontroller from STM32F4 family is crucial for research purposes because of its hardware FPU (Floating Point Unit) coprocessor which significantly accelerates floating point operations.

4.3. Software

Communication algorithms and a GUI PC program written in Python language were developed and implemented. All the tested algorithms during simulation were implemented in the microavionics module. The execution times of each controller are given in Tab. 2. The PD algorithm achieved the fastest realization times, the ANN controller was almost eight times worse, the fuzzy logic was even slower. The fuzzy logic controller with centroid defuzzification (the same as used in the simulation) where almost a half of the control cycle is occupied by a control task was the worst case. The execution time increase was caused by growing complexity of the algorithms and greater amount of operations to perform comparing to the PD controller.

Tab. 2. Comparison of the presented algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time, ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td>0.0056</td>
</tr>
<tr>
<td>Fuzzy (Centroid)</td>
<td>1.2421</td>
</tr>
<tr>
<td>Fuzzy (COG)</td>
<td>0.0740</td>
</tr>
<tr>
<td>ANN</td>
<td>0.0413</td>
</tr>
</tbody>
</table>

5. Conclusions

The comparison of the three mentioned control algorithms was performed on a simulated model and on a physical platform. It turned out that alternative control algorithms could be more robust and have better performance at the cost of small increase in the execution time. Unfortunately, the authors could not verify precisely the presented mathematical model with the developed platform to the full extent because of the problems with vibrations. Therefore only the execution times were tested in a real system.

6. References


Przemysław GąSIOR, MSc

Doctoral student in Institute of Control and Information Engineering, Poznan University of Technology. Graduated in 2014 and received a MSc diploma in Automatic Control and Robotics with major in Roboics. Now, as a member of PART research team his areas of research are control algorithms, attitude and altitude estimation and development of multirotor aerial platforms.

e-mail: przemyslaw.gasior@cie.put.poznan.pl

Marta GąSIOR, MSc

Graduated in 2014 and received a MSc diploma in Automatic Control and Robotics with major in Roboics.

e-mail: marta.gasior1990@gmail.com