ATTITUDE CONTROL OF MULTICOPTER

Radek Baránek

Doctoral Degree Programme (1), FEEC BUT

E-mail: xbaran10@stud.feec.vutbr.cz

Supervised by: František Šolc

E-mail: solc@feec.vutbr.cz

Abstract: This article is about a design of attitude controller for multicopter. The full attitude is controlled using three independent loops. Each loop consist of two nested, classical PID style, controllers. For tuning of the controller parameters the modeling approach was used. Despite the simplicity of the controller, its performance proved to be sufficient for attitude stabilization of multicopter.

Keywords: multicopter, PID controller, design, attitude, MATLAB

1. INTRODUCTION

Multicopter is generally a small aircraft with multiple rotors and each rotor has fixed-pitch blade. Typical numbers of rotors are 4, 6, and 8. By controlling the speed and consequently the thrust of each rotor, we can induce upside thrust (with respect to the copter frame) and different torques around each axis. This feature enables us to rotate the copter to any orientation and by applying the thrust to move to any direction. This type of small aircraft is widely used in many academic laboratories because of the mechanical simplicity.

The multicopters are generally unstable and therefore it is not a simple problem to control them. There are many approaches to control such a multicopter. From the simplest one based on standard PID controller [1],[2] to very complex techniques with difficult theory [3],[4]. In this work classical PID control approach is used. The tuning of controllers is done using Simulink model in MATLAB. Finally, the complete attitude control algorithm is implemented and tested on real hexacopter aircraft.

2. SIMULINK MODEL OF ONE AXIS

As mentioned above the parameters of controllers are obtained using MATLAB tools. Thus, at first the model has to be made up. Since full model of whole multicopter aircraft would be very difficult and time-consuming, only very simplified model of one axis is constructed. This simplified model reflects only one rotational degree of freedom. If we consider motions around each of the principal axes as independent, then each motion can be modeled by this model. Mechanical schematic of this model is on Figure 1, where motor is producing thrust T, which consequently produces torque M, which according to the moment of inertia determines the angular acceleration. This model is made with respect to the real platform and then almost no computation is needed to transfer the computed parameters to real platform. On the real platform, the system input is a control value sent to the motor controller, which control the speed and consequently the thrust of each rotor. The output of the real system is attitude of multicopter with respect to the earth frame. In the model of the real system, the input is also motor control value with the same range and since we have only one axis model the output is bank angle of the aircraft. The whole model consists of two main parts, model of motor and model of aircraft dynamics. Each part will be explained in the following subsections.

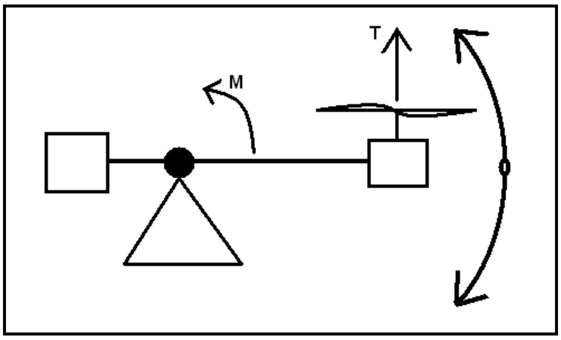

Figure 1: Simplified model of one axis.

2.1. MODEL OF MOTOR

This block of model simulates the motor with propeller. Although the effects of engine to speed of reaction seem to be negligible, it is convenient to model this phenomenon. The input to the motor model is control value which sets the current torque. This torque is summed with drag torque which is proportional to the rotor speed. The final torque is divided by moment of inertia of rotor. Then we get angular acceleration and after integration we get the rotor speed. This fact can be expressed by differential equation:

$$\frac{d\omega}{dt} = \frac{u - k\omega}{J}, \tag{1}$$

where ω is angular speed of rotor, J is moment of inertia of rotor, k is constant defining relation of control value to speed and u is control value. All in all the motor is modeled as first order static linear system. The Simulink model is on Figure 2. On Figure 3, there is a step response of motor model.

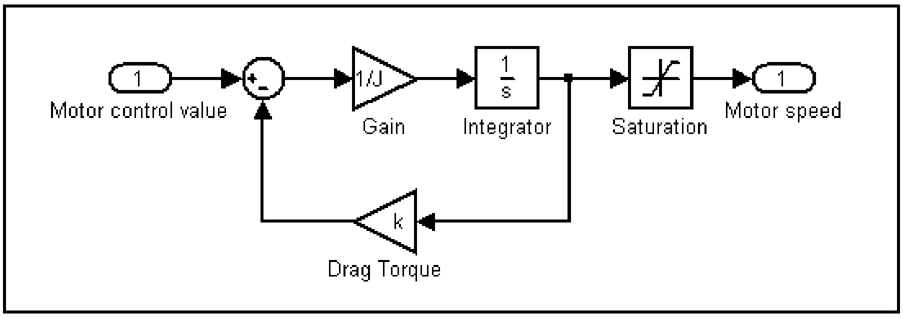

Figure 2: Simulink model of multicopter motor.

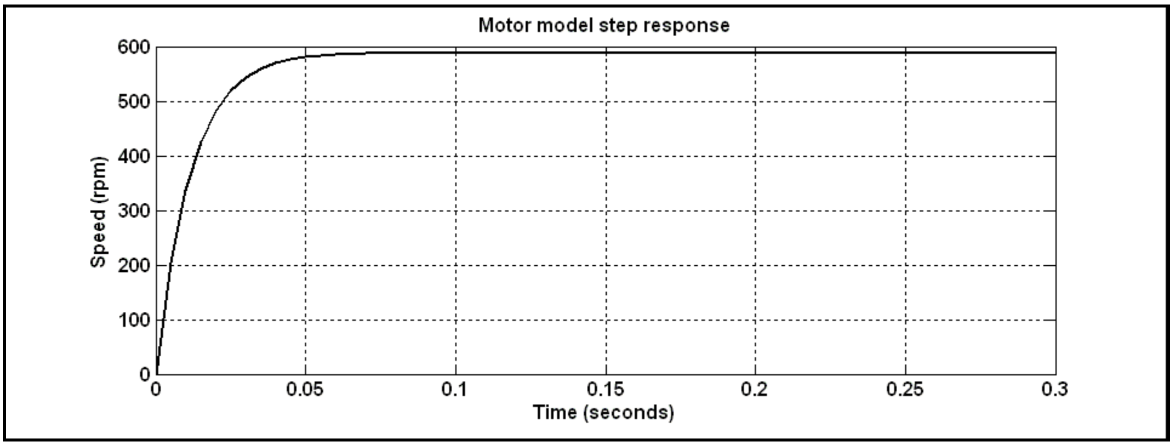

Figure 3: Step response of the motor model.

2.2. AIRCRAFT BODY DYNAMICS MODEL

Aircraft body dynamics is based on classical Newton laws. The input to the system is complete torque around specific axis generated by rotors. The output of system is angle between initial orientation and actual orientation. The input torque is divided by the moment of inertia of the copter. The resulting angular acceleration is double integrated to get the required angle. For convenience also a result of first integration is available as this is the value which gyroscopes measure on real platform. Aircraft body dynamics can be described by differential equation:

$$\frac{d^2\varphi}{dt^2} = \frac{\tau}{J_q}, \tag{2}$$

where φ is orientation angle, τ is complete torque and J_q is moment of inertia of the copter. The Simulink model of aircraft body is on Figure 4.

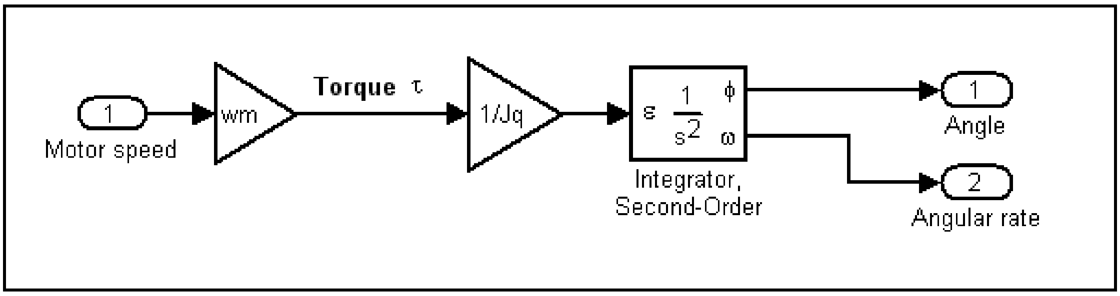

Figure 4: Simulink model of aircraft body

2.3. COMPLETE MODEL

Two above mentioned parts are combined together to make a complete one axis model of multicopter. The output of motor model is rotor speed and input of body model is complete torque, consequently output of motor model is multiplied by constant which covers number of rotors and its positions and convert rotor speed to complete torque. This relation is considered to be proportional. Since the complete model is actually third order system with two pure integrators, open loop step response has no meaning here.

3. ANGULAR RATE CONTROL LOOP

The first step to get angle controller is to control the angular rate. This is done by simple proportional (P) controller. It processes the reference angular rate and measured angular rate to output control value for motors. As the final control algorithm will be run on microcontroller, it is time-effective to use the gyroscope output value directly. This is the reason why the angular rate (rad/sec) is converted to raw gyroscopes values. This is done by multiplication of the angular rate value by constant *krg* (see Figure 5). The P constant of the controller was tuned experimentally to value P = 1/300. The Simulink model with angular rate controller is on Figure 5. The step response of angular rate controller is on Figure 6.

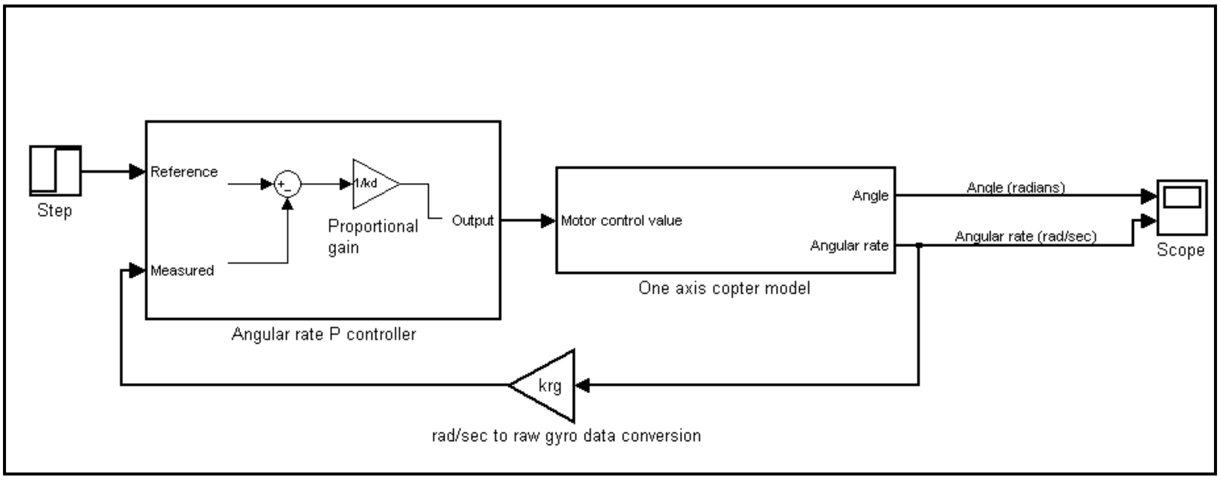

Figure 5: Simulink model of angular rate controller.

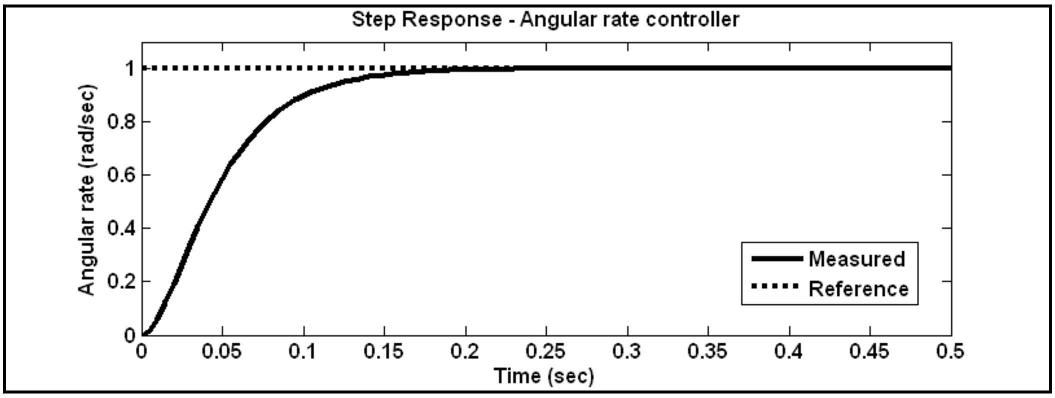

Figure 6: Step response of angular rate controller.

4. ANGLE CONTROL LOOP

For angle control PI controller was used. This controller feed the angular rate controller. For finding the best parameters for real implementation, the Simulink PID controller block was used. One of the advantages of this block is the feature to automatically tune the constants, while seeing the step response. One of the requirements for angle controller is to control with almost no overshoot. Overshoots in angle are namely uncomfortable for teleoperator. Another requirement is obviously fast response. With tuning feature of PID controller Simulink block, modeled system is automatically linearized and then user can interactively get required step response characteristic. Computed parameters can be read and also changed after the tuning process.

5. EXPERIMENT WITH REAL PLATFORM

The angle controller with parameters obtained during tuning process was tested on real hexacopter platform. Controller was implemented in its discrete form. Since during the design of model the real sensors and their ranges were taken into account, controller constants can be used directly. The algorithm of controller runs on 32-bit Cortex-M3 ARM microcontroller with sample rate of 200 Hz. The reference and measured values of roll angle grabbed during real flight are plotted in Figure 7.

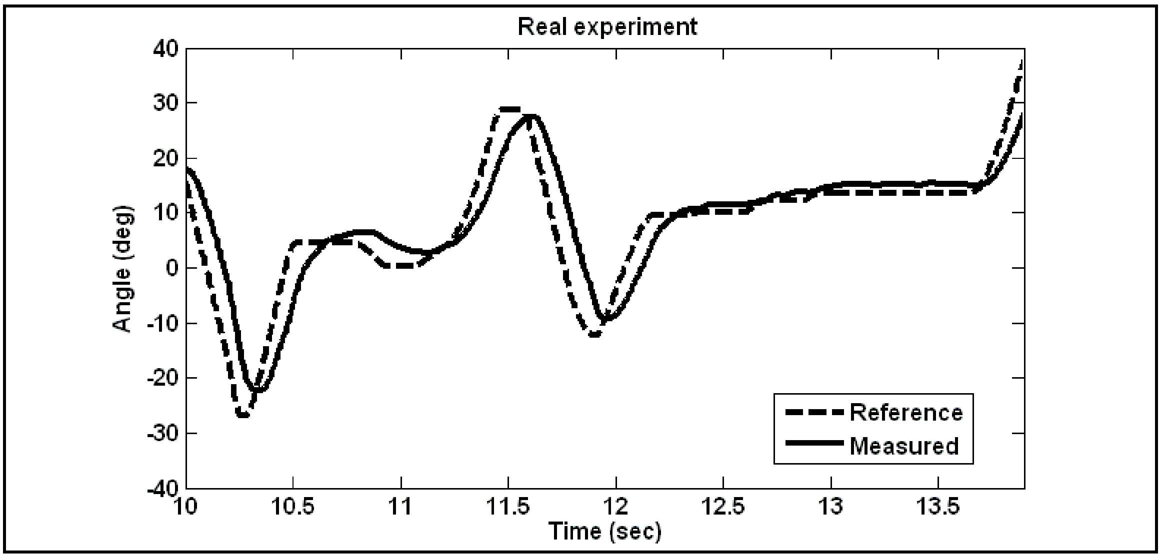

Figure 7: Data from real experiment.

6. CONCLUSION

In this article, very simple way how to control inherently unstable multicopter aircraft is presented. Despite its simplicity, Figure 6 shows sufficient performance with approximately 0.2 s delay time between reference and measured value during fast changes. Also the requirement for almost zero overshoot was met. The future work will include better orientation measurement algorithms; because it's poor quality was main limiting factor why the flight could not be fully autonomous.

ACKNOWLEDGEMENT

This work has been supported in part by Ministry of Education, Youth and Sports of the Czech Republic (Research Intent MSM0021630529 Intelligent systems in automation (VZ UAMT), Grant Agency of the Czech Republic (102/09/H081 SYNERGY - Mobile Sensoric Systems and Network) and by Brno University of Technology. Without kind support of the above-mentioned agencies and institutions the presented research and development would not be possible.

REFERENCES

[1] SALIH, A. L., M. MOGHAVVEMI, H. A. F. MOHAMED, K. S. GAEID. *Flight PID controller design for a UAV quadrotor.* Scientific Research and Essays. 2010, Vol. 5(23), s. 3660-3667. ISSN 1992-2248.

[2] MILHIM, A. B. *Modeling and Fault Tolerant PID Control of a Quad-Rotor UAV*. Canada, 2010. Master thesis. Concordia University.

[3] BOUABDALLAH, S., SIEGWART, R. *Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor.* In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain: IEEE April 2005, p. 2247-2252.

[4] DUNFIED, J., TARBOUCHI, M., LABONTE, G. *Neural Network Based Control of a Four Rotor Helicopter*. In 2004 IEEE International Conference on Industrial Technology (ICIT), Hammamet, Tunisia: IEEE, Dec. 2004, p. 1543-1548.