Ročník 2012 electroscope Číslo V

Attitude Control of Multicopter

R. Baranek [1], F. Solc[1]
[1] Department of Control and Instrumentation, Faculty of Electrical Engineering and Communication, BUT, Kolejni 4, Brno
E-mail: xbaran10@stud.feec.vutbr.cz, solc@feec.vutbr.cz

Abstract:
This article deals with the problem of attitude control of multicopters. At first the basic properties of multicopters are described. Regarding the attitude control, representation of attitude and computation of errors using rotation matrices are mentioned. Since the dynamics of rotation motion of multicopter around each axis is similar and almost independent to each other, the attitude control is split to three independent control loops. Then the construction of simplified model of multicopter with one degree of rotation freedom is described. Also the dynamics of thrusts of propellers are taken into account in the model. Finally one of the possible controller designs is described and for reference the results of controller with parameters tuned using MATLAB tool are shown.

1. INTRODUCTION

The flying robots have many useful applications in everyday life. For instance thermo vision search of missing people or inspection of fields from the skies. This branch of robotics is giving a good opportunity for testing or development of new methods in control and sensor techniques. Multicopters, mechanically very simple flying robots are typical representatives of this group.
The first step in the development of such an autonomous robot is obviously precise and robust control of attitude. There are many approaches to control attitude of the multicopter. From the simplest one based on standard PID controller [1], [2] to very complex techniques with difficult theory [3], [4]. This article describes relatively simple way of design of PID based attitude controller and parameters identification for this controller.

2. MULTICOPTERS

As mentioned above, multicopter is mechanically very simple flying robot. It consists of a multiple arms going from the center of multicopter and lying on the one plane. At the end of each arm there is a motor with propeller. All rotational axes of propellers have ideally the same direction (perpendicular to the arms plane) and therefore each propeller can produce thrust in the direction aligned with this rotational axis. Since action of force is not at the center of mass, it also generates torque as a side effect of thrust. Additionally each motor with propeller produce reactive torque as an effect of action and reaction which has opposite direction than the rotation motion of the propeller. By changing the speed of individual motors we can generate total thrust and torque and thus control the movement of the multicopter. So the only control input is array of speeds (or thrusts) of motors. The size of this array depends on the number of motors on the multicopter.
Possible multicopter constructions have at least four and also even number of propellers (arms). Only with these numbers of propellers we can fulfill the requirement of producing torque vector with arbitrary direction. If and only if this requirement is fulfilled the multicopter is fully controllable.

2.1 Generalization of the Control

Except payload, weight and size, the number of motors is the main difference between different multicopters. But every multicopter can produce only total thrust in direction perpendicular to the arm plane and torque vector with arbitrary direction (note that size of this vector is limited and depends on power of motors). From the control point of view it is convenient to use total thrust and torque vector as controlled variable against thrusts of individual propellers since these variables are not dependent on a particular multicopter and developed control algorithms can be used on all types of multicopters. For a specific multicopter direct relationship between thrusts of propellers and total thrust and torque vector can be computed in a simple way.
The procedure of computing the relationship between these two sets of variables will be shown on a general hexa-copter, the multicopter with six arms. The kinematic scheme of hexa-copter is on Fig. 1:. The computation of torques and total thrust from the thrusts of individual propeller can be easily derived using basic relation from mechanics (according to Fig. 1:):

$$\begin{bmatrix} T \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -L\frac{\sqrt{3}}{2} & -L\frac{\sqrt{3}}{2} & 0 & L\frac{\sqrt{3}}{2} & L\frac{\sqrt{3}}{2} \\ L & \frac{L}{2} & -\frac{L}{2} & -L & -\frac{L}{2} & \frac{L}{2} \\ k_{MT} & -k_{MT} & k_{MT} & -k_{MT} & k_{MT} & -k_{MT} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} \quad (1)$$

where L is a length of arms, T_x is thrust of x-th propeller (numbering of propellers according to Fig. 1:), T is a total thrust, M_i is torque around i-axis and k_{MT} is a proportional constant of relation between thrust of propeller and its reactive torque.

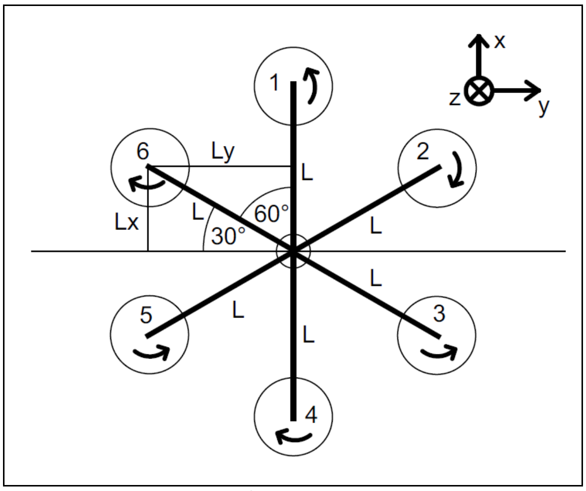

Fig. 1: Kinematic scheme of hexa-copter

But if our controller will control the total thrust and torques values, we need an inverse relation; it means formula for computing thrusts of individual propellers from the knowledge of total thrust and torques. For this the Moore-Penrose pseudo inverse can be used. Since the matrix for computation of total thrust and torque vector from individual thrusts has for any multicopter linearly independent rows, this Moore-Penrose type of pseudo inverse can be used:

$$B = A' \cdot (A \cdot A')^{-1} \qquad (2)$$

where B is pseudo inverse matrix of A, A' indicates transposition of A. Application of this pseudo inverse to the matrix from (1) yields inverse relation:

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & 0 & \frac{1}{3L} & \frac{1}{6k_{MT}} \\ \frac{1}{6} & -\frac{\sqrt{3}}{6L} & \frac{1}{6L} & -\frac{1}{6k_{MT}} \\ \frac{1}{6} & -\frac{\sqrt{3}}{6L} & -\frac{1}{6L} & \frac{1}{6k_{MT}} \\ \frac{1}{6} & 0 & -\frac{1}{3L} & -\frac{1}{6k_{MT}} \\ \frac{1}{6} & \frac{\sqrt{3}}{6L} & -\frac{1}{6L} & \frac{1}{6k_{MT}} \\ \frac{1}{6} & \frac{\sqrt{3}}{6L} & \frac{1}{6L} & -\frac{1}{6k_{MT}} \end{bmatrix} \begin{bmatrix} T \\ M_x \\ M_y \\ M_z \end{bmatrix} \qquad (3)$$

Generalized variables for control are total thrust T and torque vector $\boldsymbol{M}$ and they can be used for any general multicopter. These generalized variables are then projected using relation (3) where the size and values of matrix and number of thrusts for individual propellers will depend on the specific multicopter. This approach of generalized control brings better portability of developed control algorithms between different multicopters.

3. ATTITUDE REPRESENTATION

Multicopter is a flying vehicle generally with no position and orientation constrains. Therefore the most complex attitude representation has to be used. Rotation matrix (or direction cosine matrix, abbreviated DCM) is one of the possible representation [5].

Rotation matrix is special case of change of basis transformation matrix where basis vectors of each set are orthonormal (unit size and perpendicular to each other). Attitude information is usually coded to the rotation matrix which converts vectors from body fixed coordinate system to the reference (earth-fixed) coordinate system [6]. The elements of such a rotation matrix can be easily geometrically interpreted. The columns of rotation matrix are individual basis vector of body fixed coordinate system expressed in earth-fixed coordinate system. This fact is very useful for error computation.

3.1 Error Computation

Every standard controller of SISO (Single-Input Single-Output) system processes the error signal (the difference between reference value and measured output of system) and according to the controller algorithm compute controller output which is input to the system. Here the situation is a bit more complicated. We have reference and measured nine element matrices and controller output is a three element torque vector $\boldsymbol{M}$ only (because total thrust has ideally no impact on attitude change). But change of any element of torque vector will generally affect all elements of measured rotation matrix; therefore this MIMO (Multiple-Input Multiple-Output) system is non-linear and considerably mixed making the controller design for such a system virtually impossible. Using the geometric interpretation of rotation matrix elements simple vector calculation can be used to rapidly simplify the relation between reference and measured rotation matrices misalignment and torque vector $\boldsymbol{M}$. The goal of the computation is to determine three angles [1] which describe the misalignments between two coordinate systems (defined by measured and reference rotation matrix) and change in first element of torque vector will affect considerably only the first error angle and so on for other elements and angles. With some little vector drawing it can be shown that these computations fulfill requirements mentioned above:

$$\begin{aligned} ex &= \arcsin[(\mathbf{z}' \times \mathbf{z}) \cdot \mathbf{x}] \\ ey &= \arcsin[(\mathbf{z}' \times \mathbf{z}) \cdot \mathbf{y}] \\ ez &= \arcsin[(\mathbf{x}' \times \mathbf{x}) \cdot \mathbf{z}] \end{aligned} \qquad (4)$$

where $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ are basis vectors of body fixed frame (columns of measured rotation matrix) and $\mathbf{x}', \mathbf{y}', \mathbf{z}'$

[1] Please do not confuse these angles with Euler angles (another possible attitude representation).

are basis vectors of reference body fixed frame (columns of reference rotation matrix). With these computations the designed controller will have three error inputs and three output and we can assume that each output will be affecting just one error signal.

4. DESIGN OF THE CONTROLLER

For the purpose of design of the controller a simple model of multicopter is constructed. From previous chapter we know that we control torque vector acting on multicopter to minimize the error angles (4). We also know that first element of torque vector affects significantly only the first error angle and so on for other angles and elements. If we neglect small cross coupling effects we can split this MIMO controller into three independent control loops and each loop will act as a SISO system with controller. Next step in designing the controller is to describe the physics behind the motion of multicopter. If we consider the single loop, then the controller output will set one element of torque vector and consequently change the thrusts of individual propellers. This will generate real torque acting on a multicopter and this movement is governed by Newton's laws. This is true for each of three control loops. Therefore we can construct simplified model of rotational motion of rigid body around fixed axis and use this model for designing the controller. For full attitude control we will need three SISO controllers of the same type, but generally with different parameters.

4.1 Simplified One-dimensional Model

This model include the physics behind rotational motion as mentioned above and additionally it will reflect the finite speed of thrust changes of individual propellers. The whole model consists of two sub blocks, the model of thrust changes and the model of rotational motion.

The model of thrust changes is a simple first order linear system with one parameter J which determines the speed of changes. The greater the J parameter the slower the speed of change is. This model can be expressed by differential equation:

$$\frac{dM}{dt} = \frac{M_d - M}{J} \tag{5}$$

where M is output torque, M_d is desired (input) torque and J is the only parameter of the model. The realization of this sub block in MATLAB Simulink is on Fig. 2:.

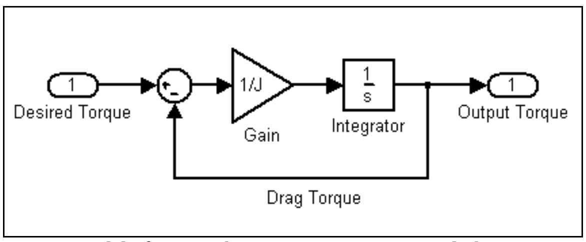

Fig. 2: Model of torque changes in MATLAB Simulink

The next sub block models the dynamics of one dimensional rotation movement according Newton's laws. The dynamics is represented by differential equation:

$$\frac{d^2\varphi}{dt^2} = \frac{M}{J_M} \tag{6}$$

where φ is output angle, M is input torque and J_M is moment of inertia of multicopter with respect to the fixed axis of rotational motion. The realization of this sub block in MATLAB Simulink is on Fig. 3:.

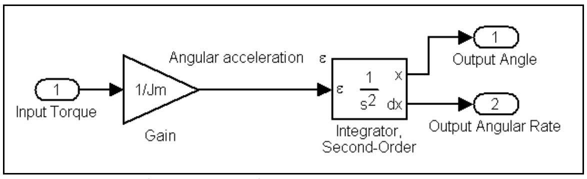

Fig. 3: Model of dynamics of rotational motion

These two sub blocks together make a complete one dimensional model of multicopter rotational movement. Since the model is known, many of standard approaches for controller design and controller parameters tuning can be used. In next subsection, one of the possible controllers is described.

4.2 Controller

Each of the three controllers is based on classic PID style controller. It consists of main loop controlling the angle and one nested loop for angular rate control. The angular rate controller is a simple proportional controller which processes the error between reference angular rate and measured angular rate. For the purpose of measuring the attitude multicopters are most often equipped with gyroscopes, accelerometers and magnetometers. The attitude is then computed by advanced algorithms. Since the gyroscope measure the angular rate directly, this value is available for angular rate controller without the need for a discrete differentiation. Theoretically, according to one dimensional model, this controller cannot be over gained. But with higher values of proportional term we let the noise from gyroscope to pass to the thrusts of propellers, which can leads to shaking and instability of the multicopter. The optimal value of proportional term of angular rate controller depends on the quality of used gyroscope. Main reason for using nested angular rate controller is that it lowers the order of controlled system and thus avoids using the D term in parent angle controller. The D term is namely problematic in general PID controllers.

For main angle control the classic PI controller is used. It is fed by the angle error and outputs the desired angular rate which is input to the nested angular rate controller. The whole scheme of controller and model is on Fig. 4:. Note that in real controller implementation angle error computation is substituted by computations from (4).

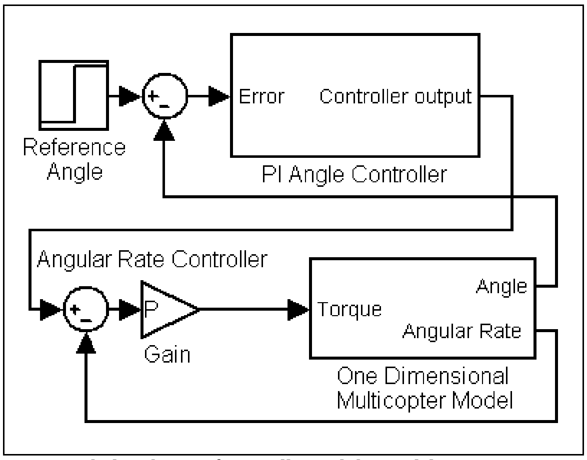

Fig. 4: Whole scheme of controller and the model in MATLAB Simulink

For full attitude control we need three controllers of this type. Each controller can have different parameters depending on multicopter characteristics (moment of inertia etc.). For the identification of parameters of the controller many methods can be used. The proportional gain of angular rate can be determined experimentally, but for angle PI controller this is not recommended since badly tuned angle controller can lead to the multicopter damage.

5. EXPERIMANTAL RESULTS

This controller was tested on real hexa-copter platform. The controller algorithm run on the 32-bit microcontroller and the attitude was measured using complementary filter algorithm which processes data from tri-axis gyroscope, accelerometer and magnetometer. The angular rate controller was tuned experimentally. The value of proportional value was the same for all three controllers:

$$P_{RATE} = 0.5 \tag{7}$$

For tuning the parameters of main PI angle controllers the auto tuning tool of MALAB Simulink was used. For this purpose the identification of model parameters was performed. The values are in following table:

Model parameters according to real hexa-copter

Variable	***Sub block***	***Value***
J	Model of thrust changes	0.005 s^{-1}
J_{MX}	Model of	0.0123 Kgm^2
J_{MY}	rotation	0.0144 Kgm^2
J_{MZ}	movement	0.0288 Kgm^2

As the values of moment of inertia around different axis differ, also the controller parameters will be different. Tuned parameters of controllers are summarized in the following table:

Controller parameters obtained using MATLAB auto tune

Term	***x axis***	***y axis***	***z axis***
P_{RATE}	0.5	0.5	0.5
P_{ANGLE}	21.5	22.7	29.1
I_{ANGLE}	7.5	8	10

Step response of the controller for the x axis simulated using MATLAB is on Fig. 5:.

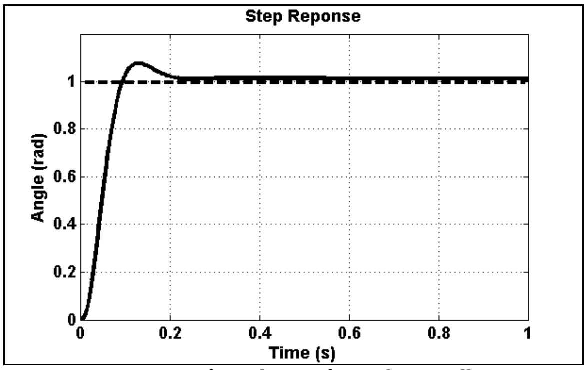

Fig. 5: Step response of simulation of attitude controller

As you can see on Fig. 5: the designed controller performs well in simulation. As we use generalized variables to control we also have to know all parameters for inverse calculation (3). These parameters are summarized in the next table:

Parameters for inverse thrust calculation

Variable	***Value***
L	0.2 m
k_{MT}	0.12 Nm/N

The drivers of motors on a real hexa-copter are not controlled by thrust values but by control byte (0-255) it depends on used propeller what thrust is generated. For our hexa-copter the relation was almost linear with proportional constant:

$$Tb = 0.015\,N \tag{8}$$

The corresponding control value for motor driver is computed from relation:

$$C_{VALUE} = \frac{T_x}{Tb} \tag{9}$$

where T_x is motor thrust. It is difficult to measure step response in real flight condition because this measurements needs a lot of free space, so for illustration of controller performance only data captured during hard maneuvers are plotted in Fig. 6: On the plot you can see approximately 0.2 s delay between reference and measured value. This delay sufficiently corresponds to the simulated step response data where the step time is also approximately 0.2 s.

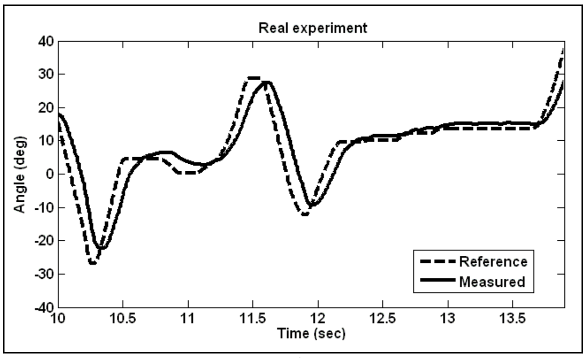

Fig. 6: Data captured during real flight experiment with hexa-copter

6. CONCLUSION

This article presents a relatively simple solution of attitude control of multicopters. For control the generalized variables defined in chapter 2 was used and this concept proved to be appropriate as attitude control algorithm designed for generalized variables can be ported to any multicopter. But there still remain parameters like controller parameters or parameters for inverse computation of thrusts where values are strongly dependent on specific multicopter design. Even the attitude controller based on elementary PID controller has very sufficient performance in simulation and even in real flight (Fig. 5:, Fig. 6:). Future work will certainly include better approach of tuning of parameters of controller as the MATLAB auto tune feature is not well transparent and we have no idea what theory states behind this tuning method.

ACKNOWLEDGMENT

This work was supported by grant “(Supporting) Research of Modern Methods and Approaches in Automation” from the Internal Grant Agency of Brno University of Technology (grant No. FEKT-S-11-6).

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 Ap-plied 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.

[5] Chaturvedi, N.A., Sanyal, A.K., McClamroch, N.H., "Rigid-Body Attitude Control," Control Systems, IEEE, vol. 31, pp. 30-51, 2011.

[6] Titterton, D.H., Weston, J.L. *Strapdown inertial navigation technology (second edition).* Massachusetts, Paul Zarchan, Editor-in-chief. 2004, 558 p.