



## STATE SPACE CONTROL SYSTEM FOR RECONNAISSANCE FLYING ROBOT WITH ROTOR NONLINEARITY SUPPRESSION

KRIZ, V[lastimil] & ZALUD, L[udek]

**Abstract:** This paper describes the design of control system for reconnaissance flying robot. The nonlinear mathematical model of the robot for control system design is divided into two parts. The first part involves the nonlinearity of rotors and the second part the nonlinearity of system dynamics. The nonlinearity of the rotors is then suppressed by creating block with reverse transfer function, and the second part is linearized around operating point. This allows effective use of the whole range of rotors speed together with using simple linear state space control system.

**Key words:** copters, control system, stabilization, state-space

### 1. INTRODUCTION

The goal is to create control system for reconnaissance flying robot. This robot will carry a camera or other necessary sensors for reconnaissance missions. For this purpose quadrotor was chosen as a best platform (Albers et al., 2010). The advantage of this platform is its robustness and ability to hover. The disadvantage is its inherent instability. Because of this those robots require control systems for stabilizing.

### 2. CONTROL SYSTEM DESIGN

Developed system is based on already existing platform (Gabrlík, 2010). But for reconnaissance mission, where carrying a camera is required, the stability of this quadrotor was not good. Because of this, new control system was proposed.

The original control system in platform was based on PD controllers. Data for this controller was provided from inertial navigation unit. To utilize all system variables, which are this unit able to provide (Bradac, 2011), the state-space control system was designed.

#### 2.1 Mathematical model

Definitions of the frames and variables are shown in Fig.1. Symbol assignment is following:  $\omega$  – angular speed,  $v$  – translation speed,  $x, y, z$  – position,  $\phi, \theta, \psi$  – roll, pitch and yaw angles,  $n$  – rotation speed of rotor.

Because the system is nonlinear, the classic approach is to linearize the system around operating point and design controller based on this linearized model (Solc, 2009), or to create more complex controller which is able to handle this nonlinearity (Bouabdallah et al., 2007).

The proposed approach is divide the model into two parts: The first part will contains the nonlinear relations between rotors speed and forces and torques affecting the robot, and the second part the nonlinearity of system dynamics. This allows simpler form of controller and better dynamics of rotors.

Input to the first part will be rotation speed of rotors and the output will be fictive variables  $u_1 - u_4$ . The meaning of those variables is following:  $u_1$  – torque in  $x$  axis caused by different thrust of rotors 2 and 4,  $u_2$  – torque in  $y$  axis caused by different thrust of rotors 1 and 3,  $u_3$  – torque in  $z$  axis caused by different

reaction torque from rotors rotating in opposite direction,  $u_4$  – force in  $z$  axis caused by common thrust of all rotors.

Those variables are inputs to the second part of model and the outputs will be the position, angles, and rotational and translation speeds of the robot.

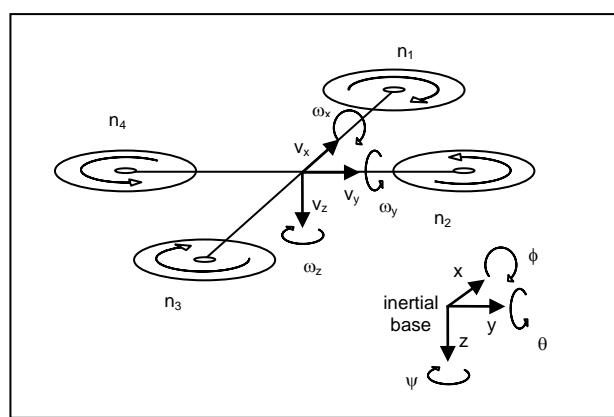


Fig.1. Definition of variables and bases

The first part of model can be described by static equations:

$$u_1 = (-n_2^2 + n_4^2)k_T l \quad (1)$$

$$u_2 = (n_1^2 - n_3^2)k_T l$$

$$u_3 = (-n_1 + n_2 - n_3 + n_4)k_M$$

$$u_4 = (n_1^2 + n_2^2 - n_3^2 + n_4^2)k_T$$

Where the  $k_T$  is constant between rotor thrust and quadrotor of rotation speed of rotor,  $k_M$  is constant between rotor reaction torque and quadrotor of rotation speed of rotor, and  $l$  is the distance between the centre of the robot and motors.

The second part of model consists of 4 sets of equations describing the robot dynamics: Equations for rotational movement, equations for translation movement, equations for transformation angular and translation speeds between frames, and equations for transformation gravity force to robot axis components.

Rotational movement:

$$\dot{\omega}_x = \frac{u_1 + (I_{yy} - I_{zz})\omega_y\omega_z + u_1}{I_{xx}} \quad (2)$$

$$\dot{\omega}_y = \frac{u_2 + (I_{zz} - I_{xx})\omega_x\omega_z + u_2}{I_{yy}}$$

$$\dot{\omega}_z = \frac{u_3 + (I_{xx} - I_{yy})\omega_x\omega_y + u_3}{I_{zz}}$$

Symbol  $I$  in these equations represents moment of inertia around defined axis.

Translation movement:

$$\begin{aligned}\dot{v}_x &= \frac{m\omega_y v_z + m\omega_z v_y + G_x}{m} \\ \dot{v}_y &= \frac{m\omega_x v_z + m\omega_z v_x + G_y}{m} \\ \dot{v}_z &= \frac{m\omega_x v_y + m\omega_y v_x + G_z - u_4}{m}\end{aligned}\quad (3)$$

Transformation between frames (Solc, 2009):

$$\begin{aligned}\dot{\phi} &= \omega_x + \omega_y \sin \phi \operatorname{tg} \theta + \omega_z \cos \phi \operatorname{tg} \theta \\ \dot{\theta} &= \omega_y \cos \theta - \omega_z \sin \theta \\ \dot{\psi} &= \omega_y \frac{\sin \phi}{\cos \theta} + \omega_z \frac{\cos \phi}{\cos \theta} \\ \dot{x} &= v_x (\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) \\ &\quad - v_y (\cos \phi \sin \psi - \cos \psi \sin \phi \sin \theta) \\ &\quad + v_z \cos \theta \cos \psi \\ \dot{y} &= v_y (\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi) \\ &\quad - v_z (\cos \psi \sin \phi - \cos \phi \sin \theta \sin \psi) \\ &\quad + v_x \cos \theta \sin \psi \\ \dot{z} &= v_z \cos \phi \cos \theta - v_x \sin \theta + v_y \cos \theta \sin \phi\end{aligned}\quad (4)$$

Gravity force in robot axes:

$$\begin{aligned}G_x &= -mg \sin \theta \\ G_y &= mg \cos \theta \sin \phi \\ G_z &= mg \cos \theta \cos \phi\end{aligned}\quad (5)$$

After substituting  $G_x$ ,  $G_y$ , and  $G_z$  from equations 5 in equations 3, then equations 2, 3, and 4 form state space description of the system.

## 2.2 Linearization and rotor nonlinearity suppression

Because of extracting the nonlinearity of rotors to separate (first) part of model, the second part of model contains the system dynamic where the only nonlinearity is the trigonometric functions of angular speeds and angles. Because for intended mission is it expected that the quadrotor will be operating only in horizontal position we can linearize those equations around this operating point ( $\phi, \theta, \omega_x, \omega_y = 0$ ).

The principle of suppression the rotor nonlinearity rests in creation of block, which will have the reverse transfer function than the first part of model. This block together with the first part of model will have unitary transfer function. The principle is shown in Fig.2.

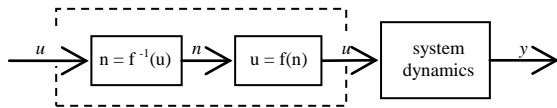


Fig.2. Rotor nonlinearity suppression

The block can be described by following equations:

$$\begin{aligned}n_1 &= \sqrt{\frac{2k_M u_2 - k_T l u_3 + k_M l u_4}{4k_M k_T l}} \\ n_2 &= \sqrt{\frac{-2k_M u_1 + k_T l u_3 + k_M l u_4}{4k_M k_T l}} \\ n_3 &= \sqrt{\frac{-2k_M u_2 - k_T l u_3 + k_M l u_4}{4k_M k_T l}}\end{aligned}\quad (6)$$

$$n_4 = \sqrt{\frac{2k_M u_1 + k_T l u_3 + k_M l u_4}{4k_M k_T l}}$$

## 3. CONCLUSION

This approach allows us to describe the system in linear form, which leads to simple control system.

We also do not need to linearize the first part of model around operating point (some nominal  $n$  of rotors) and use only limited scale of  $n$ , but we can utilize the whole possible range of rotors speed. This enable good dynamic of copter.

The state space controller was computed by method of pole placement from the linearized form of equations 2, 3, and 4. The step response of the robot with designed control system to transit from  $\theta = -50$  to  $\theta = 0$  in time 1 sec. is shown in Fig.3.

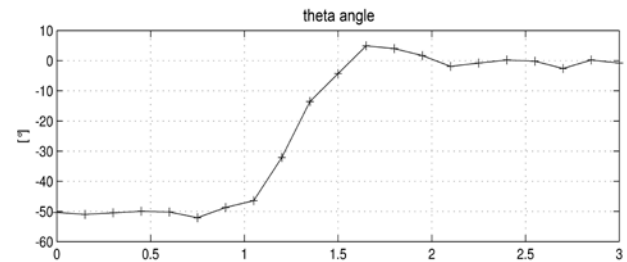


Fig.3. Step response of the robot in one axis

The test flights show that the control system is able to stabilize well the quadcopter in horizontal position. The problem is to keep position, because the inertial navigation unit does not provide precise data of the position. Thus the next research will be how to accurately estimate position of the copetr. This expects equip the robot with additional sensors and data fusion from these sensors. The goal is to have copter, which is able autonomously hold position.

## 4. ACKNOWLEDGEMENTS

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

## 5. REFERENCES

- Albers, A.; Sander, C. & Kindermann, M. (2010). Development of a Lightweight Unmanned Aerial Vehicle for Rescue and Maintenance Tasks, *Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium*, ISSN 1726-9679, ISBN 978-3-901509-73-5, Katalinic, B. (Ed.), pp. 0925-0926, DAAAM International Vienna, Vienna
- Bouabdallah, S.; Siegwart, R. (2007). *Design and control of quadrotors with application to autonomous flying*, EPFL, Lausanne, pp. 054-075
- Bradac, F. (2011). *Quadrocopter - sensory subsystem*. Brno University of technology, Faculty of Electrical Engineering and Communication, Brno
- Gabrlík, P. (2010). *Quadrocopter - modeling, stabilization, control*. Brno University of technology, Faculty of Electrical Engineering and Communication, Brno
- Solc, F. (2009). Derivation of equations of motion, Euler kinematic equations, mathematical model of quadrotor, Available from: [sftp://dzin1.feec.vutbr.cz/app/everyone/vyuka/SOLC/quadrotor/quadrotor\\_all.pdf](sftp://dzin1.feec.vutbr.cz/app/everyone/vyuka/SOLC/quadrotor/quadrotor_all.pdf) Accessed: 2010-11-03