Stabilization of a Quadcopter

Alejandro Alonso García
Department of electronic systems
Control and Automation
Aalborg University
Email: aalons16@student.aau.dk

Amalie V. Petersen
Department of electronic systems
Control and Automation
Aalborg University
Email: @student.aau.dk

Andrea Victoria Tram Løvemærke
Department of electronic systems
Control and Automation
Aalborg University
Email: @student.aau.dk

Niels Skov Vestergaard
Department of electronic systems
Control and Automation
Aalborg University
Email: nveste12@student.aau.dk

Noelia Villarmarzo Arruñada
Department of electronic systems
Control and Automation
Aalborg University
Email: nvilla16@student.aau.dk

***Abstract*—Quadcopters are becoming increasingly interesting due to the great variety of usage. In this paper the aim is to design a system, that will make the quadcopter fly stably and steadily. The system's coupled behavior and instability raises a challenging control task. We will solve this task by implementing a controller design, that is based upon a model that is derived by first principles of physics. This is later linearized by the Taylor approximation. The controller is made up of multiple subsystems. An attitude controller and a translational controller are designed as state space control and classical control respectively. The prototype does not carry on board sensor, but gets its sensor input from Vicon and the control is done in the micro processor on the quadcopter. This is a distributed system, where delays must be considered in order to obtain optimal control.**

I. Introduction

In the last years, the interest for quadcopters has increased due to the great possibilities they offer. Among these, the most well-known ones are surveillance, inspection of big structures and search and rescue missions in difficult environments **droneuses**

The quadcopter constitutes a control challenge due to its unstable nature and coupled behavior. The system has 6 degrees of freedom, the 3 position coordinates and the 3 orientations, and there are only four actuation variables which are the motor velocities. The dimension of the problem is explained by McKerrow in **draganflyer**

The control of a quadcopter has been addressed many times in the recent years. In Mian et al. **backstepping** the quadcopter is controlled using a back-stepping technique and nonlinear controllers. Other way of solving the issue is presented in Tayebi et al. **quaternionsPD** in which the quadcopter attitude is modeled using quaternions and controlled with a PD based controller. In **MianWang** Mian and Yang model the system using its dynamic equations and use non linear controllers to achieve a steady flight while in Mokhtari et al. **GHinf** the system is controlled by a mixture of a robust feedback linearizion and a linear GH_∞.

The approach presented here models the quadcopter by a first principles method. This approach yields a non linear model that describes the attitude and translational behavior of the quadcopter. The model is then linearized around an equilibrium point, which is chosen to be in hovering position. With the linearized equations, controllers for attitude and translational behaviors are designed. The angular controller is obtained by means of a State Space representation while the translational controller is designed using classical control techniques. In the control system, the translational constitutes an outer loop and sets the reference for the attitude controller.

In the last part of the paper, the simulations and experimental results of the designed controllers are shown and discussed.

II. Model

The quadcopter system is shown in Figure 1. As it can bee seen, the system is modeled by using two coordinate frames. The inertial frame is utilized to describe the translational movement while the body frame is attached to the quadcopter and used to characterize its attitude behavior. In the figure, also the positive references for rotational and translational movements are depicted, as well as the main forces and torques acting on the quadcopter.

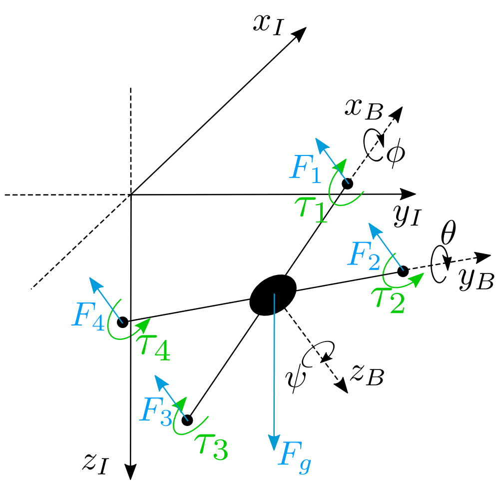

Figure 1: Quadcopter diagram showing the forces and torques acting on the system and the positive references chosen for rotations and translations in both Inertial and Body coordinate frames.

The forces generated in the propeller are easily explained in the Body coordinate frame. In order to represent them in the Inertial frame a rotation matrix is used. It is built considering a 123 rotation sequence **rotationmatrix**

The dynamic model of the quadcopter can be explain through three sets of equations. The first describes the motor and the propeller, the second presents the attitude response of the quadcopter and the third explains hos the translational variables of the system evolve with time.

A. Motor and Propeller

The four motors in the quadcopter generate the rotation required so the propeller creates the force that lifts the quadcopter. This force is called thrust force and can be modeled as proportional to the square of the motor rotational velocity. MAYBE SOURCE. The coefficient for this equation is called thrust coefficient and has been found by experiments. The rotation of the propellers also generates a torque on each motor due to drag between air and propeller. Drag torque is compensated in the quadcopter by having two of the motors turning in one direction and the two others in the opposite. It can also be described as proportional to the square of the velocity by terms of a drag coefficient that has also been obtained through tests. Equation 1 and 2 show the expression for the thrust force and drag torque caused by the rotation of the propeller.

$$F = k_{th}\omega^2 \tag{1}$$
$$\tau = k_d\omega^2 \tag{2}$$

Where:

F	is the thrust force	[N]
k_{th}	is the thrust coefficient	[N s^2 rad^{-2}]
ω	is the angular velocity	[rads$^-1$]
τ	is the drag torque	[Nm]
k_d	is the drag coefficient	[N m s^2 rad^{-2}]

This equations are included in the attitude and translational models derived below.

B. Attitude Model

The attitude model equations are based on Newton Second Law for rotational movement and are represented in Equation 3, 4 and 5.

$$J_x\ddot{\phi} = k_{th}(\omega_4^2 - \omega_2^2)L \tag{3}$$
$$J_y\ddot{\theta} = k_{th}(\omega_1^2 - \omega_3^2)L \tag{4}$$
$$J_z\ddot{\psi} = k_d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \tag{5}$$

INCLUDE WHERE

The expressions above state how the thrust force difference between motors 1 and 3 affects the roll angular acceleration, how that between motors 4 and 2 affects the pitch angle and how the yaw acceleration depends on the four motors by means of the drag torque generated on the propeller.

C. Translational Model

The equations describing the response of the system along the x, y and z axes is derived from Newton's Second Law. The forces that act on the system are those from the propellers and the gravitational. These expressions are shown in Equation 6, 7 and 8. FIX EQUATIONS

$$\mathrm{m\ddot{x}_I = -k_{th}}({\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2 + {\omega_4}^2)\sin(\theta) \tag{6}$$
$$\mathrm{m\ddot{y}_I = -k_{th}}({\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2 + {\omega_4}^2)(-\sin(\phi))\cdot\cos(\theta) \tag{7}$$
$$\mathrm{m\ddot{z}_I = F_g - k_{th}}({\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2 + {\omega_4}^2)\cos(\phi)\cdot\cos(\theta) \tag{8}$$

It is worth mentioning that, as the thrust forces always point in the negative z direction in the body coordinate frame, the accelerations in x and y directions in the inertial frame are zero as long as pitch and roll angles are 0.

III. Linearization

The linearization of the model equations has been developed following the first order Taylor approximation around an equilibrium point of the system. The chosen point is the hovering position and that implies that all variables have a value of zero, that is, the translational and attitude accelerations, velocities and positions. Choosing a zero acceleration equilibrium point along the Inertial z axis yields a equilibrium rotational speeds so that the necessary thrust is generated to compensate for the gravitational force.

The resulting equations for the attitude model after the linearization are shown in Equation 9, 10 and 11.

$$J_x\Delta\ddot{\phi} = 2k_{th}L(\varpi_4\Delta\omega_2 - \varpi_2\Delta\omega_4) \tag{9}$$
$$J_y\Delta\ddot{\theta} = 2k_{th}L(\varpi_1\Delta\omega_1 - \varpi_3\Delta\omega_3) \tag{10}$$
$$J_z\Delta\ddot{\psi} = 2k_d(\varpi_1\Delta\omega_1 - \varpi_2\Delta\omega_2 + \varpi_3\Delta\omega_3 - \varpi_4\Delta\omega_4) \tag{11}$$

INCLUDEWHERE Similarly, the equations of the translational model are linearized. The result is shown in 12, 13 and 14. **FIX EQUATIONS**

$$m \cdot \Delta \ddot{x}_I = -k_{th}(\overline{\omega}_1{}^2 + \overline{\omega}_2{}^2 + \overline{\omega}_3{}^2 + \overline{\omega}_4{}^2)\cos(\overline{\theta})\Delta\theta \tag{12}$$

$$m \cdot \Delta \ddot{y}_I = k_{th}(\overline{\omega}_1{}^2 + \overline{\omega}_2{}^2 + \overline{\omega}_3{}^2 + \overline{\omega}_4{}^2)\cos(\overline{\phi})\cos(\overline{\theta})\Delta\phi \tag{13}$$

$$m\Delta \ddot{z}_I = -2\ k_{th}(\overline{\omega}_1\Delta\omega_1 + \overline{\omega}_2\Delta\omega_2 + \overline{\omega}_3\Delta\omega_3 + \overline{\omega}_4\Delta\omega_4)\cos(\overline{\phi})\cos(\overline{\theta}) \tag{14}$$

INCLUDEWHERE

IV. Control

The control of the system is divided into two control systems. One handles the attitude and the other controls the translational behavior of the quadcopter.

A. Attitude Controller

The attitude controller for the quadcopter has been designed using a state space representation of the system. This helps handling the coupled angular response of the quadcopter. The chosen states for the system are the three angular positions and the three angular velocities. The input vector of the attitude system consists of the four motor rotational speeds and the output vector consists of the three angles, roll, pitch and yaw. Below, the state, input and the output vectors are presented.

$$\mathbf{x}(t) = \begin{bmatrix} \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$$

$$\mathbf{y}(t) = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$$

$$\mathbf{u}(t) = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T$$

The above is then used in construction of the state space matrix representation as displayed in Equation 15 and 16.

$$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{u}(t) \tag{15}$$

$$\mathbf{y}(t) = \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot \mathbf{u}(t) \tag{16}$$

Where:

- **A** is the system matrix
- **B** is the input matrix
- **C** is the output matrix
- **D** is the feed forward matrix

The values for the A, B, C and D matrices are obtained from the linearized attitude equations, yielding the matrices shown below. As D is a zero matrix, only A, B and C are shown.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{2 \cdot \mathrm{k_{th}} \cdot \mathrm{L} \cdot \overline{\omega}_2}{\mathrm{J_x}} & 0 & \frac{2 \cdot \mathrm{k_{th}} \cdot \mathrm{L} \cdot \overline{\omega}_4}{\mathrm{J_x}} \\ \frac{2 \cdot \mathrm{k_{th}} \cdot \mathrm{L} \cdot \overline{\omega}_1}{\mathrm{J_y}} & 0 & -\frac{2 \cdot \mathrm{k_{th}} \cdot \mathrm{L} \cdot \overline{\omega}_3}{\mathrm{J_y}} & 0 \\ \frac{2 \cdot k_d \cdot \overline{\omega}_1}{J_z} & -\frac{2 \cdot k_d \cdot \overline{\omega}_2}{J_z} & \frac{2 \cdot k_d \cdot \overline{\omega}_3}{J_z} & -\frac{2 \cdot k_d \cdot \overline{\omega}_4}{J_z} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

State feedback and integral control For the quadcopter to hover, it is desired to keep the attitude in equilibrium; this is achieved using a state feedback. To be able to change and track a reference an integral controller is designed around the state feedback. This allows changing the attitude in order to move in the xy-plane of the inertial system.

B. Translational Controller

The movements of the quadcopter along the inertial frame directions x, y and z are controlled by the translational controllers. It is decided to structure the controllers as cascade loops. The relation between the controllers are presented in [1]

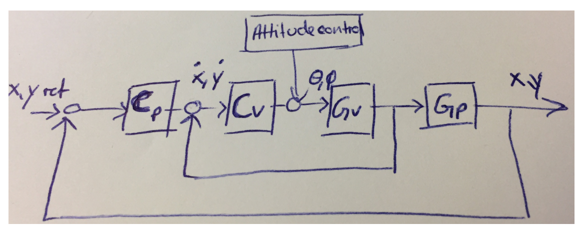

Figure 2: Cascade loop for the translational controllers where the inner and outer loop control velocity and position respectively.

The vertical position is controlled by the cascade loop for the z axis' velocity and position, obtaining the required sum of motor rotational speeds. The x and y controller share similar properties as the output for each are an angle reference, namely θ_{ref} and ϕ_{ref} respectively. Firstly the x and y controllers are designed similarly followed by an individual design for the z controller.

The inner loop for the x and y translational controllers are now designed followed by the outer loop. The model equations derived previously, see Equation 6 and 7, are Laplace transformed and put on transfer function in respect to the inner loop, yielding:

$$G_{\dot{x}_I}(s) = \frac{\dot{x}_I(s)}{\theta(s)} = \frac{-k_{th}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)}{m\ s} \tag{17}$$

$$G_{\dot{y}_I}(s) = \frac{\dot{y}_I(s)}{\phi(s)} = \frac{k_{th}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)}{m\ s} \tag{18}$$

[1] FiXme Note: figure

Where:

G_{x_I} is the plant for the translational velocity in x_I direction [1]

G_{y_I} is the plant for the translational velocity in y_I direction [1]

The plants are similar but with different signs. The controller design is carried out for the x translational velocity and applied to the y translational velocity afterwards.
A proportional controller is sufficient as the plant already has an integrator, that will eliminate a steady state error and output disturbances. The gain will be the same for both controllers, but must be negative for the x translational controller in order to compensate for its negative plant as this will otherwise be unstable in the closed loop.

The final proportional controllers are as follows:

$$C_{\dot{x}I(s)=-0.19} \tag{19}$$

$$C_{\dot{y}I(s)=0.19} \tag{20}$$

Where:

G_{x_I} is the plant for the translational position in x_I direction [1]

G_{y_I} is the plant for the translational position in y_I direction [1]

The gain is designed such that it encounters a bandwidth that is three times lower than the attitude model to ensure minimum effect of disturbances. The plant of the outer loop is simply an integrator to transform velocity to position. The controller of the outer loop is a proportional controller. The outer loop is designed to have three times less bandwidth than the inner loop to ensure minimization of disturbances to secure a stable system. The proportional controllers are the following:

$$C_{\dot{x}O(s)=XX} \tag{21}$$

$$C_{\dot{y}0(s)=YY} \tag{22}$$

WHEREs for the controllers are not needed i think - take up too much space. The translational z controller is designed as cascade as well. The

V. Results

Simulation vs. reality.
Comment on the results and how that correlates with reality, without discussing possible issues or improvements.

VI. Discussion

Discussing possible issues or improvements of the above results.

VII. Conclusion

Summary - what we want the reader to remember.

VIII. Acknowledgments

Henrik Schiøler , associated professor at Aalborg University
Christoffer Sloth, associated professor at Aalborg University