Attitude and Position Control of a Quadcopter using a Networked Distributed System

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: alavem13@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, a design that is able to make the quadcopter hover and move to a desired position is presented. The system's coupled behavior and instability raises a challenging control task. This task is solved by implementing a controller design, that is based upon a model that is derived by first principle physics. This is later linearized using the Taylor approximation, since it is desired to use a linear approach in the controllers. The total system is made up of multiple subsystems. An attitude and a translational controller are designed as state space control and classical control respectively. The prototype does not carry on board sensors, but gets its position and orientation from a remote sensor, keeping the control in a micro processor on the quadcopter. This layout constitutes a distributed system, where network issues such delays and packet losses need to be taken into account.**

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 rotational speeds. 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 non-linear 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. Since the sensors are not placed in the quadcopter and the information comes from an external motion tracking system **vicon** an analysis on how the network could affect the control loop is also presented. 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.

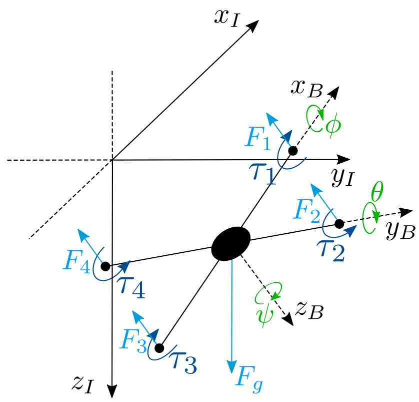

Figure 1: Quadcopter diagram showing the forces and torques acting on the system and the positive references chosen for rotations and translations in both Inertial and Body coordinate frames.

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.

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 how the translational variables of the system evolve.

A. Motor and Propeller

The four motors in the quadcopter generate a rotation in the propellers that 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. The coefficient for this equation is called thrust coefficient and has been found through experiments. This rotation 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, k_{th} is the thrust coefficient, ω is the angular speed of the motor, τ is the drag torque and k_d is the drag coefficient.

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

B. Attitude Model

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

$$J_x\ddot{\phi} = k_{th}(\omega_4^2 - \omega_2^2)L \tag{3}$$

$$J_y\ddot{\theta} = k_{th}(\omega_1^2 - \omega_3^2)L \tag{4}$$

$$J_z\ddot{\psi} = k_d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \tag{5}$$

Where J_x, J_y and J_z are the moments of inertia around the three axis of rotation, $\ddot{\phi}$, $\ddot{\theta}$ and $\ddot{\psi}$ are the accelerations in roll, pitch and yaw angles respectively, ω_i is the rotational speed of each motor and L is the distance between the center of the quadcopter and the position of the motors.

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 torques generated on the propellers.

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 one. These expressions are shown in Equation 6, 7 and 8.

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

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

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

Where m is the mass of the quadcopter, $\ddot{x}$, $\ddot{y}$ and $\ddot{z}$ are the accelerations in the directions of the inertial reference frame, ϕ, θ and ψ are the roll, pitch and yaw angles respectively and F_g if the gravitational force.

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.

D. 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.

$$J_x \ \Delta\ddot{\phi} = 2 \ k_{th} \ L \ \overline{\omega}_4 \ \Delta\omega_4 \ - \ 2 \ k_{th} \ L \ \overline{\omega}_2 \ \Delta\omega_2 \tag{9}$$

$$J_y \ \Delta\ddot{\theta} = 2 \ k_{th} \ L \ \overline{\omega}_1 \ \Delta\omega_1 \ - \ 2 \ k_{th} \ L \ \overline{\omega}_3 \ \Delta\omega_3 \tag{10}$$

$$\begin{aligned} J_z \ \Delta\ddot{\psi} = {} & 2 \ k_d \ \overline{\omega}_1 \ \Delta\omega_1 \ - \ 2 \ k_d \ \overline{\omega}_2 \ \Delta\omega_2 \\ & + \ 2 \ k_d \ \overline{\omega}_3 \ \Delta\omega_3 \ - \ 2 \ k_d \ \overline{\omega}_4 \ \Delta\omega_4 \end{aligned} \tag{11}$$

Where $\Delta\ddot{\phi}$, $\Delta\ddot{\theta}$ and $\Delta\ddot{\psi}$ are the changes in rotational acceleration from equilibrium, $\overline{\omega}_i$ is the rotational speed of each motor in equilibrium and $\Delta\omega_i$ is the change in rotational speed of each motor from equilibrium.

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

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

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

$$\begin{aligned} m \ \Delta\ddot{z}_I = {} & - 2 \ k_{th} \ \overline{\omega}_1 \ \Delta\omega_1 - 2 \ k_{th} \ \overline{\omega}_2 \ \Delta\omega_2 \\ & - 2 \ k_{th} \ \overline{\omega}_3 \ \Delta\omega_3 - 2 \ k_{th} \ \overline{\omega}_4 \ \Delta\omega_4 \end{aligned} \tag{14}$$

Where $\Delta\ddot{x_I}$, $\Delta\ddot{y_I}$ and $\Delta\ddot{z_I}$ are the changes in linear acceleration from equilibrium in each direction of the inertial frame and $\Delta\phi$ and $\Delta\theta$ are the changes in roll and pitch from equilibrium respectively.

III. 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 $\mathbf{A}$ is the system matrix, $\mathbf{B}$ is the input matrix, $\mathbf{C}$ is the output matrix and $\mathbf{D}$ is the feedback matrix.

The values for the $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ matrices are obtained from the linearized attitude equations, yielding the matrices shown below. As $\mathbf{D}$ is a zero matrix, only $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ 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 Figure 2.

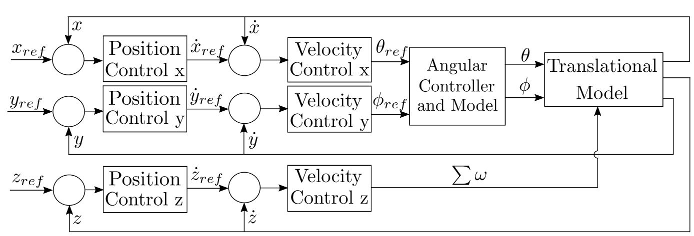

Figure 2: Overview of control structure.

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:

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

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

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 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 inner loop for the z translational controller is first designed followed by the outer loop. The model equation derived previously, see Equation 8 is Laplace transformed and put on transfer function, where the velocity in z direction is the output and the sum of motor rotational speeds is the input.

$$G_{\dot{z}} = \frac{\dot{z}}{\omega_{sum}} = \frac{\frac{1}{4}\ (-2k_{th})\ \overline{\omega}_{sum}}{m\ s} \quad (19)$$

Where $\dot{z}_I$ is the velocity in the z_I direction, ω_{sum} is the sum of the rotational speeds of the motors and $\overline{\omega}_{sum}$ is the sum of the rotational speeds in equilibriumm.

Due to an integrator and a negative gain the system's locus will move into the right half plane as the gain increases. A proportional controller with negative gain will ensure the system to become stable. However, due to input disturbances an integrator is added, as this will eliminate this issue. The final z translational controller of the inner loop is a PI-controller.

Before the designed controllers can be implemented on the micro controller on the quadcopter, they must be discretized. The discretization is done as billinear approximation, also known as Tustin, where the half plane side in Laplace domain is mapped as the unit cycle in the discrete domain.

IV. Network

The effects on the control performance of the usage of the wireless network to get the sensor data can be divided in two. These are the delay and the packet loss.

The theoretical modeling of these effects has been studied by many researchers in order to obtain stability criteria and maximum allowable delay or packet dropout. However, this approach often leads to an increased complexity in the model [1].

An alternative to account for these effects in the control system is using a network simulator like TrueTime **TrueTime** This toolbox helps finding the maximum delay and packet loss probability such that the control system is still stable by simulating the controllers and the network together.

The delay is modeled as an exponential distribution whose mean parameter has been obtained experimentally by averaging multiple delay measurements in the communication channel.

The packet loss is defined as a constant probability of loosing a packet in the network. This probability is also obtained experimentally by [2].

V. Results

Simulation vs. reality.

Symbol	Value	Units
m	0.996	kg
L	0.225	m
J_x	0.01073	$kg\ m^2$
J_y	0.01073	$kg\ m^2$
J_z	0.02135	$kg\ m^2$
k_{th}	1.32922 10^{-5}	$N\ s^2\ rad^{-2}$
k_d	9.39741 10^{-7}	$N\ m\ s^2\ rad^{-2}$
$\overline{\omega}_i$	429	$rad\ s^{-1}$

Table I: Parameters used in the simulation of the model.

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

Model parameters: [3]

Controllers:

The inner loop proportional controller of the x and y translational cascade controllers are $C_{\dot{x}}(s) = -0.19$, $C_{\dot{y}}(s) = 0.19$ and the outer loops are $C_x(s) = 0.55$, $C_y(s) = 0.55$ The inner loop proportional controller of the z translational cascade controller is $C_{\dot{z}}(s) = \frac{-201s+0.8}{s}$ and the outer loop is $C_z = TBD$ [4].

From this the following simulation and test are derived:

VI. Discussion

Discussing possible issues or improvements of the above results.

VII. Conclusion

In this project, the behavior of a quadcopter has been modeled by first principles of physics. A control system has been designed in order to hover and move to a desired position.

[1]FiXme Note: THEORETICAL APPROACHES SOURCES
[2]FiXme Note: HOW TO FINISH THIS
[3]FiXme Note: bring table with values
[4]FiXme Note: to do

The control system has been split into an attitude and a translational controller. The first one has been designed using a state space approach, including state feedback with integral control and a reduced order observer. The translational control system has been designed with a classical control approach and result in three cascade loops, including proportional and PI controllers. As the quadcopter uses an external motion tracking system to determine its position and orientation, an analysis of the issues that can arise when having a networked distributed system has been done in order to ensure the control system remains stable.

VIII. Future Work

Several actions can be done in order to optimize this prototype. The most significant is to implement on board sensors such that the quadcopter will not be limited to only operate within the Vicon room. Hereafter there is a wide range of potential applications that can be enforced.

IX. Acknowledgments

[5]

[5] FiXme Note: commented out until final version is to be handed in