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: apet13@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.

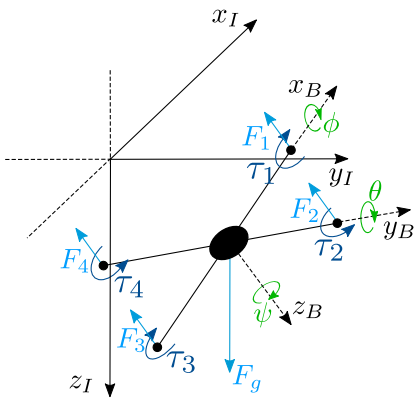

Figure 1: Quadcopter diagram showing the forces and torques acting on the system and the positive references chosen for rotations and translations in both Inertial and Body coordinate frames.

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.

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

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

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

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.

$$\overline{\omega}_i = \sqrt{\frac{m\ g}{4k_{th}}} \tag{9}$$

The resulting equations for the attitude model after the linearization are shown in Equation 10, 11 and 12.

$$J_x\ \Delta\ddot{\phi} = 2\ k_{th}\ L\ \overline{\omega}_4\ \Delta\omega_4\ -\ 2\ k_{th}\ L\ \overline{\omega}_2\ \Delta\omega_2 \tag{10}$$

$$J_y\ \Delta\ddot{\theta} = 2\ k_{th}\ L\ \overline{\omega}_1\ \Delta\omega_1\ -\ 2\ k_{th}\ L\ \overline{\omega}_3\ \Delta\omega_3 \tag{11}$$

$$J_z\ \Delta\ddot{\psi} = 2\ k_d\ \overline{\omega}_1\ \Delta\omega_1\ -\ 2\ k_d\ \overline{\omega}_2\ \Delta\omega_2 + 2\ k_d\ \overline{\omega}_3\ \Delta\omega_3\ -\ 2\ k_d\ \overline{\omega}_4\ \Delta\omega_4 \tag{12}$$

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 Equation 13, 14 and 15.

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

$$m\ \Delta\ddot{y}_I = k_{th}\ (\overline{\omega}_1{}^2 + \overline{\omega}_2{}^2 + \overline{\omega}_3{}^2 + \overline{\omega}_4{}^2)\ \Delta\phi \tag{14}$$

$$m\ \Delta\ddot{z}_I = -\ 2\ k_{th}\ \overline{\omega}_1\ \Delta\omega_1 - 2\ k_{th}\ \overline{\omega}_2\ \Delta\omega_2 - 2\ k_{th}\ \overline{\omega}_3\ \Delta\omega_3 - 2\ k_{th}\ \overline{\omega}_4\ \Delta\omega_4 \tag{15}$$

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. This two are related such that the translational one sets the references for the angles that the inner one needs to control.

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 16 and 17.

$$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{u}(t) \tag{16}$$

$$\mathbf{y}(t) = \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot \mathbf{u}(t) \tag{17}$$

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.

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

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

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

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 combined with the other one. It is also desirable to use an observer in order to estimate the angular velocities that are part of the state of the system. This is done using a reduced-order observer. These two designs can be done independently due to the separation principle. **ssReference**

Figure 2 shows how this designs are related.

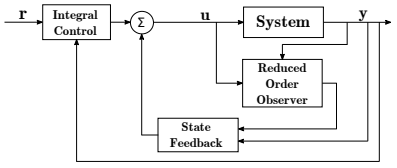

Figure 2

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 them as cascade loops, where the velocity and position are controlled in the inner and outer loop respectively. The relation between the controllers is presented in Figure 3.

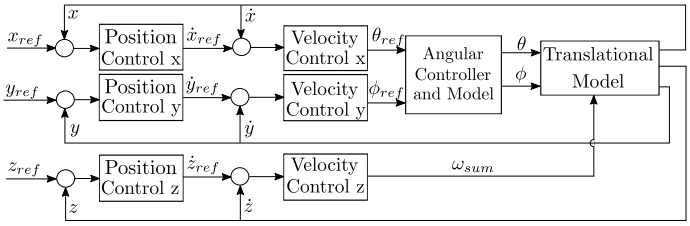

Figure 3: Overview of control structure.

The x and y controller share similar properties as the output for each are an angle reference, θ_{ref} and ϕ_{ref} respectively, while the output of the z controller is the required sum of rotational speeds in the motors. Firstly the x and y controllers are designed, followed by an individual design for the z controller.

To design the inner controllers for x and y velocities, the model equations derived previously, see Equation 6 and 7, are Laplace transformed. They are used to create a transfer function between the angles and the velocities, yielding:

$$G_{\dot{x}}(s) = \frac{\dot{x}(s)}{\theta(s)} = \frac{-k_{th}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)}{m\ s} \tag{18}$$

$$G_{\dot{y}}(s) = \frac{\dot{y}(s)}{\phi(s)} = \frac{k_{th}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)}{m\ s} \tag{19}$$

Where G_{x_I} and G_{y_I} are the plants used to design the velocity controllers in x_I and y_I direction respectively.

It can be noticed that the plants are the same but with different signs. That is why the controller design is carried out for the x translational velocity and applied to the y translational velocity afterwards.

A proportional controller is considered the plant already has an integrator, that eliminates a steady state error and output disturbances. The gain is 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 **bandwidthReference**

The plant of the outer loop is simply an integrator to transform velocity to position. As in the inner loop, the controller of the outer loop is a proportional controller. The outer loop is again designed to have three times less bandwidth than the inner loop to ensure minimization of disturbances and a stable system.

To be able to design the inner loop for the z translational controller, the model equation derived previously, see Equation 8 is Laplace transformed and written on the form of a transfer function, being the velocity in z direction is the output and the sum of motor rotational speeds is the input.

$$G_{\dot{z}} = \frac{\dot{z}}{\omega_{sum}} = \frac{\frac{1}{4}\ (-2k_{th})\ \overline{\omega}_{sum}}{m\ s} \tag{20}$$

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 equilibrium.

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 and to have zero steady state error. However, there are input disturbances given as the equilibrium rotational speeds of the motors maybe be not exactly equal in reality than the calculated ones. In this case, an integrator in the plant is not enough to handle them so another one is added to the controller to eliminate this issue. The final z translational velocity controller results in a PI-controller.

As in the case of the x and y outer controllers for position, the plant is just an integrator and, since the disturbances are handled by the inner one, a proportional controller is considered sufficient. This is as well designed so that the bandwidth frequency is three times lower than the inner one.

IV. Network

The wireless network between the sensor and the quadcopter can influence the performance of the controller. This influence can be divide into to categories: the delay, from when the sensor receives data to the microcontroller utilizes the data, and the loss of packages in the communication channel.

The theoretical modeling of these influences has been studied by several researchers in order to find a maximum allowable delay and packet loss while the control system keeps stable [1]. However, this approach often leads to an increased complexity in the model.

An alternative to account for these effects in the control system is to utilize a network simulator like TrueTime **TrueTimeNew**

TrueTime enables the opportunity to simulate the network and the controller together. It is thereby possible to find the probability for packet loss and the maximum occurring delay in the system while insuring the control system is still stable. This can hereafter be compared to the delay occurring in reality. To find the maximum delay in reality, the delay is modeled as an exponential distribution with a mean parameter obtained experimentally by averaging multiple delay measurements in the communication channel. The packet loss, defined as a constant probability of loosing a packet, is found experimentally by sending a large amount of packages and see how many of these packages are lost.

V. Results

The controllers have been simulated in MATLAB Simulink in order to generate results presented in this section.

The model parameters can be seen in Table I. All of them have been obtained through test but the moments of inertia, which were calculated with following an analytical procedure.

[1] FiXme Note: THEORETICAL APPROACHES SOURCES

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 though the analysis and design.

The attitude controller is defined chosen feedback and integral poles and the observer poles. These are chosen to be $[-6 - 6.2 - 6.4 - 6.6 - 6.8 - 7 - 7.2 - 7.4 - 7.6]$ and $[-20 - 2530]$ respectively.

The translation velocity controllers of the x and y translational are $C_{\dot{x}}(s) = -0.1$, $C_{\dot{y}}(s) = 0.1$ and the position ones are $C_x(s) = 0.5$, $C_y(s) = 0.5$. 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 $C_z = 0.9$

This controllers have been discretized using th Tustin method with a sampling period of 30 ms.

Regarding the network, the mean delay selected is 30 ms and the package loss probability has been set to 0 due to the low sending frequency used in the system. The simulation results obtained with these parameters and controllers are shown in Figure 4.

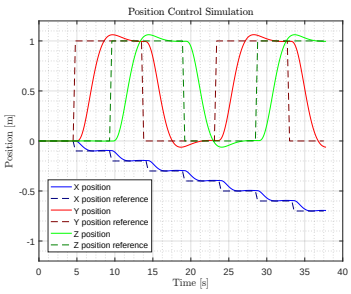

Figure 4: Position control results in the three inertial axes directions. The references given to the control system are shown with dashed lines.

The inner attitude controller results are also included in and shown in Figure 5 so the performance of the attitude control can be evaluated.

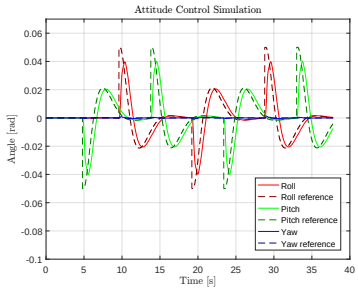

Figure 5: Attitude control results in the three angles. The references given to the attitude control system are shown with dashed lines.

VI. Discussion

The results obtained in the simulations show both the position and the angular response of the quadcopter.

It can be seen from them that the controllers achieve their goal even though the network delay and the sampling rate affect the performance. The main network effect is the designed bandwidth of the controllers. This occurs due to the limited frequency in which the sensor data is obtained from the motion tracking system.

It is also worth observing how the angular controllers show a permanent error with respect to the reference. This is generated as the integral control has been designed, for the sake of simplicity, as having constant reference applied. This consideration does not affect the final position of the quadcopter.

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. 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.

[2]

[2] FiXme Note: Acknowledgements