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

Alejandro Alonso García, Amalie Vistoft Petersen, Andrea Victoria Tram Løvemærke,
Niels Skov Vestergaard and Noelia Villarmarzo Arruñada
Department of Electronic Systems
Automation and Control
Aalborg University
Email: [aalons16] [apet13] [alavem13] [nveste12] [nvilla16] @student.aau.dk

***Abstract*—Quadcopters are becoming increasingly interesting due to the great variety of usage. 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 linear control design, which is based upon a model derived by first principles modeling. The control system is divided into attitude and translational. These are designed using state space and classical control methods, respectively. The quadcopter gets its attitude and position from an external motion tracking system based on infrared cameras, while the control remains in the microcontroller on the quadcopter. This layout constitutes a distributed system, where network issues occur. These are considered in the control design to ensure the stability.**

I. Introduction

In the last years, the interest for quadcopters has increased due to the multiple possibilities they offer. Among these are search and rescue missions in difficult environments, inspection of big structures and surveillance. [1]

The quadcopter constitutes a control challenge due to its unstable nature and coupled behavior. It has six degrees of freedom, which are three angles and three position coordinates. To control these, there are four actuation variables, namely the motor rotational speeds. [2]

The control of a quadcopter has been addressed multiple times in recent years. In Mian et al. [3] the quadcopter is controlled using a back-stepping technique and non-linear controllers. Another way of solving this control task is presented in Tayebi et al. [4], here the quadcopter's attitude is modeled using quaternions and controlled with a PD based controller. In [5], Mian and Wang model the system using its dynamic equations and design non-linear controllers to achieve a steady flight, while in Mokhtari et al. [6], the system is controlled by a mixture of a robust feedback linearization and a modified optimization control method.

This paper examines the achievable performance using a state space design strategy for the attitude control combined with a cascaded structure with classical linear translational controllers. Furthermore, the effects of the attitude controller's bandwidth on the translational controllers are considered along with the effect of remote sensing on the system.

In section II, the model of the quadcopter is obtained by first principles modeling. 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 the hovering position. Controllers for attitude and translational behaviors are designed in section IV.Since the sensors are not placed on the quadcopter and the data comes from an external motion tracking system [7], an analysis on how the network effects can be considered when designing the controllers is also given. This is described in section III. In section V, the simulations and experimental results of the designed controllers are provided and hereafter they are discussed in section VI. Lastly, a conclusion is presented in section VII and possible future work is mentioned in section VIII.

II. Model

A free body diagram of the quadcopter is seen in Figure 1.

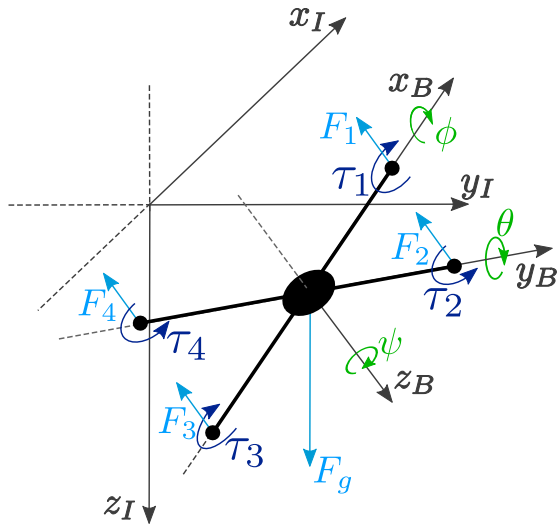

Figure 1: Forces ($\mathbf{F}_i$) and torques ($\boldsymbol{\tau}_i$) acting on the quadcopter, positive references for rotations and positive references for translations in both the inertial and body coordinate frames.

As seen in the figure, 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 describe its attitude behavior. The positive references for rotational and translational movements

are depicted in Figure 1, as well as the main forces and torques acting on the quadcopter. The positive rotations follow the right-hand rule.

The forces generated in the propellers are obtained 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 [8]. This means that any rotation is described as three rotations first around the x_{B} axis, then around the y_{B} axis and lastly around the z_{B} axis.

The dynamic model of the quadcopter is given by three sets of equations, which are presented in this section. The first describes the motor and the propeller, the second and the third characterize the attitude and translational behavior of the quadcopter, respectively.

A. Motor and Propeller

The four motors on the quadcopter generate a rotation in the propellers, which creates the force that lifts the quadcopter. This thrust force is modeled proportional to the square of the motor rotational speed. The thrust coefficient for each motor is found experimentally.

The rotation also generates a torque on each motor due to the aerodynamic drag. It is also described as proportional to the square of the rotational speed in terms of a drag coefficient, which is also obtained experimentally. The total drag torque is compensated in the quadcopter by having two of the motors turning in one direction and the two others in the opposite direction.

The expressions for the thrust force and drag torque caused by the rotation of each propeller are

$$F = k_{\mathrm{th}}\omega^2 \ , \tag{1}$$

$$\tau = k_{\mathrm{d}}\omega^2 \ , \tag{2}$$

where F is the thrust force, k_{th} [$\mathrm{Ns^2rad^{-2}}$] is the thrust coefficient, ω is the angular speed of the motor, τ is the drag torque and k_{d} [$\mathrm{Nms^2rad^{-2}}$] is the drag coefficient.

B. Attitude Model

The attitude model equations, based on Newton's Second Law for rotational movement, state how the thrust forces and the drag torques in (1) and (2) influence the attitude behavior:

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

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

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

where J_x, J_y and J_z are the moments of inertia around the three axes of rotation. $\ddot{\phi}$, $\ddot{\theta}$ and $\ddot{\psi}$ are the angular accelerations in roll, pitch and yaw. ω_i is the rotational speed of each motor. L is the distance between the center of the quadcopter and the position of the motors. The parameters of the model are found in section V.

C. Translational Model

The equations describing the response of the system along the inertial x, y and z axes are derived from Newton's Second Law of Motion. The forces acting on the system are the thrust forces and the gravitational force. By using (1)

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

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

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

where m is the mass of the quadcopter, $\ddot{x}_{\mathrm{I}}$, $\ddot{y}_{\mathrm{I}}$ and $\ddot{z}_{\mathrm{I}}$ are the accelerations along the inertial reference frame directions, ϕ, θ and ψ are the roll, pitch and yaw angles and F_g is the gravitational force acting on the quadcopter. The parameters of the model are found in section V.

D. Linearization

The model equations are linearized using the first order Taylor approximation around an equilibrium point, which is the hovering position. This implies that the attitude and translational accelerations and velocities are zero. In addition, the attitude is set to zero.

Choosing a zero acceleration linearization point along the z_{I} axis yields an equilibrium rotational speed in each motor. These four rotational speeds generate a thrust force that counteracts the gravitational force. This is expressed as

$$\overline{\omega}_i = \sqrt{\frac{mg}{4k_{\mathrm{th}}}} \ . \tag{9}$$

The resulting equations for the attitude model after the linearization are

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

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

$$\begin{aligned} J_z\Delta\ddot{\psi} =& 2k_{\mathrm{d}}\overline{\omega}_1\Delta\omega_1 - 2k_{\mathrm{d}}\overline{\omega}_2\Delta\omega_2 \\ & + \ 2 \ k_{\mathrm{d}}\overline{\omega}_3\Delta\omega_3 - 2 \ k_{\mathrm{d}}\overline{\omega}_4\Delta\omega_4 \ , \end{aligned} \tag{12}$$

where $\Delta\ddot{\phi}$, $\Delta\ddot{\theta}$ and $\Delta\ddot{\psi}$ are the changes in rotational acceleration from the linearization point, $\overline{\omega}_i$ is the rotational speed of each motor to achieve equilibrium along the z_{I} axis and $\Delta\omega_i$ is the change in rotational speed of each motor from the linearization point.

Similarly, the equations of the translational model are linearized. The result is

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

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

$$\begin{aligned} m\Delta\ddot{z}_{\mathrm{I}} =& - 2k_{\mathrm{th}}\overline{\omega}_1\Delta\omega_1 - 2k_{\mathrm{th}}\overline{\omega}_2\Delta\omega_2 \\ & - 2k_{\mathrm{th}}\overline{\omega}_3\Delta\omega_3 - 2k_{\mathrm{th}}\overline{\omega}_4\Delta\omega_4 \ , \end{aligned} \tag{15}$$

where $\Delta\ddot{x}_{\mathrm{I}}$, $\Delta\ddot{y}_{\mathrm{I}}$ and $\Delta\ddot{z}_{\mathrm{I}}$ are the changes in linear acceleration from the linearization point in each direction of the inertial frame and $\Delta\phi$ and $\Delta\theta$ are the changes in roll and pitch from the linearization point.

III. Network

The wireless network between the external sensor and the quadcopter influences the performance of the controller. This influence is considered and two effects, delay and missed packets, are examined.

The theoretical modeling of these influences has been studied by several researchers with the purpose of understanding how the stability of the control system is affected when a network is used [9], [10]. However, these approaches often lead to an increase in the complexity as the network effect is taken into account in the model of the system.

Stability of the system when influenced by the network is instead analyzed using the network simulator TrueTime [11]. It provides the ability to simulate the network model, the controller design and the system model together. This approach makes it possible to design the controllers taking into account the network effects and, thus, ensuring that stability is achieved.

The delay is modeled in the network simulation as constant for all samples. Its value is calculated by adding two time intervals. The first is the time needed for the transmission of the data, that is, the time elapsed since the data is acquired until it is available for the controller. This is a fixed delay formed by a combination of transmission and code execution times. The second is the maximum time until the controller utilizes the data. It is estimated as the time between two consecutive packets. This yields the maximum delay, thereby considering the worst case scenario.

The missed packets, defined as a constant probability of the controller using old data, is found experimentally by sending a large amount of packets and examining how many control loops run with old data.

IV. Control

The design is divided into two control systems. One handles the attitude and the other the translational behavior. These two are related such that the translational controller sets the references for the angles handled by the attitude controller. When designing them, the influences of the network are taken into account.

A. Attitude Controller

The attitude controller is 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 consists of the four motor rotational speeds and the output vector consists of the three angles, roll, pitch and yaw.

The state, input and output vectors are

$$\mathbf{x}(t) = \begin{bmatrix} \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T,$$
$$\mathbf{y}(t) = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T,$$
$$\mathbf{u}(t) = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T.$$

The values for the $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ matrices in the state space model are obtained from the linearized attitude equations (10), (11) and (12), yielding

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

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

Note, that the $\mathbf{D}$ matrix is a zero matrix.

The attitude control is based on a state feedback and an integral term in order to track a given reference and reject input disturbances. As not all states are measured a reduced order observer is implemented to estimate the angular velocities. Due to the separation principle, both subsystems can be designed independently. Figure 2 shows how these designs are related. [12]

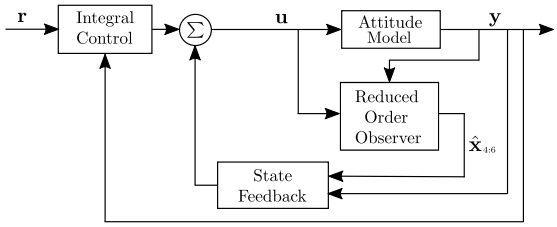

Figure 2: Control structure for the system, including the state feedback with integral action and the reduced order observer.

The design of the state feedback with the integral action is shown in Figure 3.

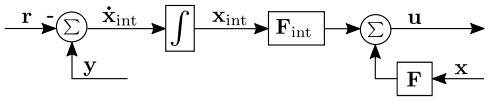

Figure 3: State feedback with integral action in the attitude control structure, where $\mathbf{r}$ is the angular reference vector and $\mathbf{x}_{\text{int}}$ contains the integral states.

Three states, $\mathbf{x}_{\text{int}}$, are added to the already existing state vector in order to track the two references for ϕ and θ and

reject input disturbances in the three angles. This leads to the extended system

$$\dot{\mathbf{x}}_\mathrm{e} = \mathbf{A}_\mathrm{e}\mathbf{x}_\mathrm{e} + \mathbf{B}_\mathrm{e}\mathbf{u} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix} \mathbf{r}, \tag{16}$$

$$\mathbf{y} = \mathbf{C}_\mathrm{e}\mathbf{x}_\mathrm{e}, \tag{17}$$

where

$$\dot{\mathbf{x}}_\mathrm{e} = \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_\mathrm{int} \end{bmatrix}, \qquad \mathbf{A}_\mathrm{e} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{B}_\mathrm{e} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}, \qquad \mathbf{C}_\mathrm{e} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix}.$$

The feedback law is given by

$$\mathbf{u} = \mathbf{F}_\mathrm{e}\mathbf{x}_\mathrm{e} = \mathbf{F}\mathbf{x} + \mathbf{F}_\mathrm{int}\mathbf{x}_\mathrm{int} \ . \tag{18}$$

The feedback matrix, $\mathbf{F_e}$, is obtained by using a Linear Quadratic Regulator (LQR) approach together with Bryson's rule, [12]. This makes it possible to find the optimal feedback gains while being able to prioritize the different states and set boundaries for the control action. LQR design is based on minimizing the following cost function,

$$J = \int_0^\infty \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u} \ dt \ . \tag{19}$$

The matrices $\mathbf{Q}$ and $\mathbf{R}$ must be selected as positive semi-definite and positive definite matrices, respectively. These penalize the error of the different states and inputs of the system.

These matrices are selected by using Bryson's rule, which defines $\mathbf{Q}$ and $\mathbf{R}$ as diagonal matrices whose elements are

$$Q_{ii} = \frac{1}{\text{maximum acceptable value of } [x_i^2]} \ , \tag{20}$$

$$R_{ii} = \frac{1}{\text{maximum acceptable value of } [u_i^2]} \ . \tag{21}$$

The values are chosen to ensure stability of the controller while the states converge to the equilibrium point and the references are tracked as fast as possible. This process is performed using the simulation of the model including network effects. These values are then used to find the optimal feedback matrices, which minimize the cost function (19) as

$$\mathbf{F}_\mathrm{e} = -\mathbf{R}^{-1}{\mathbf{B}_\mathrm{e}}^T\mathbf{P} \ , \tag{22}$$

where the matrix $\mathbf{P}$ is positive definite and its value is found using the Algebraic Riccatti Equation,

$$\mathbf{A}_\mathrm{e}^T\mathbf{P} + \mathbf{P}\mathbf{A}_\mathrm{e} - \mathbf{P}\mathbf{B_e}\mathbf{R}^{-1}\mathbf{B}_\mathrm{e}^T\mathbf{P} + \mathbf{Q} = \mathbf{0} \ . \tag{23}$$

Once $\mathbf{F}_\mathrm{e}$ is obtained, it is split into $\mathbf{F}$ and $\mathbf{F}_\mathrm{int}$. In this way, the controller is implemented as shown in Figure 3.

The reduced order observer estimates the angular velocities by means of the system input and output. With this approach, the first three states, $\mathbf{x}_{1:3}$, are equal to the outputs, $\mathbf{y}$, whereas the other three states, $\mathbf{x}_{4:6}$, are estimated as $\hat{\mathbf{x}}_{4:6}$.

The observer is designed by finding the matrix $\mathbf{L}_\mathrm{obs}$ such that the eigenvalues of the matrix $\mathbf{A}_{22}+\mathbf{L}_\mathrm{obs}\mathbf{A}_{12}$ have negative real part. This makes the estimate converge to the true angular velocity [12]. This is performed by splitting the original system matrices into the submatrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \ .$$

With the observer matrix, the observer equation is derived, see (24). This ensures an estimate $\hat{\mathbf{x}}_{4:6}$ which converges to $\mathbf{x}_{4:6}$ at a rate given by the chosen observer poles.

$$\begin{aligned} \dot{\hat{\mathbf{x}}}_{\mathbf{4:6}} + \mathbf{L}_\mathrm{obs}\dot{y} = (\mathbf{A}_{22} + \mathbf{L}_\mathrm{obs}\mathbf{A}_{12})\hat{\mathbf{x}}_{4:6} \\ + (\mathbf{A}_{21} + \mathbf{L}_\mathrm{obs}\mathbf{A}_{11})\mathbf{y} + (\mathbf{B}_2 + \mathbf{L}_\mathrm{obs}\mathbf{B}_1)\mathbf{u} \end{aligned} \tag{24}$$

The estimation of $\mathbf{x}_{4:6}$ is implemented as shown in Figure 4.

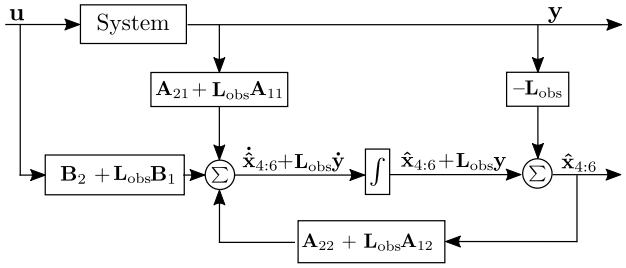

Figure 4: Detailed diagram of the reduced order observer illustrating its implementation.

B. Translational Controller

The translational controllers follow a cascade structure, where the velocity is the inner loop and the position is the outer loop. The relation between the controllers is presented in Figure 5.

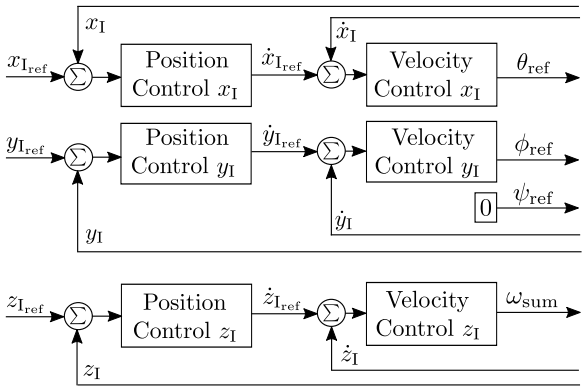

Figure 5: Overview of the translational controllers structure. $\overline{\omega}_\mathrm{sum}$ is the sum of rotational speeds requested by the z_I controller.

The x_I and y_I controllers share similar properties as both their outputs are angle references, θ_ref and ϕ_ref. The translational movement in x_I and y_I can be obtained by only changing these angles as long as ψ is zero. For this reason, ψ_ref is set to

zero at all times. The output of the z_I controller is the required sum of the motor rotational speeds.

In order to design the inner controllers for the velocities $\dot{x}_\mathrm{I}$, $\dot{y}_\mathrm{I}$ and $\dot{z}_\mathrm{I}$, the linear equations derived previously, see Equation 13, 14 and 15, are Laplace transformed. These are used to create the transfer functions

$$G_{\dot{x}} = \frac{\dot{x}_\mathrm{I}}{\theta} = \frac{-k_\mathrm{th}(\overline{\omega}_1^2 + \overline{\omega}_2^2 + \overline{\omega}_3^2 + \overline{\omega}_4^2)}{ms} \ , \tag{25}$$

$$G_{\dot{y}} = \frac{\dot{y}_\mathrm{I}}{\phi} = \frac{k_\mathrm{th}(\overline{\omega}_1^2 + \overline{\omega}_2^2 + \overline{\omega}_3^2 + \overline{\omega}_4^2)}{ms} \ , \tag{26}$$

$$G_{\dot{z}} = \frac{\dot{z}_\mathrm{I}}{\omega_\mathrm{sum}} = \frac{-2k_\mathrm{th}\overline{\omega}_\mathrm{sum}}{4ms} \ , \tag{27}$$

The three plants contain an integrator that can handle steady state errors and output disturbances. However, if there are input disturbances, such as different motor characteristics or external causes like wind, an integrator term in the controller is needed. Then, a zero is added to eliminate the marginal stability due to the presence of only two integrators in the system. Finally, the gain is adjusted to reduce the oscillations. It is worth mentioning that, since the plants for the x_I and z_I velocities have a negative gain, the controller needs to compensate for it with a negative gain.

The bandwidths of the $\dot{x}_\mathrm{I}$ and $\dot{y}_\mathrm{I}$ controllers are designed to be three times lower than that of the inner attitude controller. This reduces the influence of the attitude behavior on the velocity controller. The attitude controller has a bandwidth of 2 rad·s^{-1}, yielding a bandwidth of 0.7 rad·s^{-1} for these two velocity controllers.

The plants of the position control loops contain only an integrator, which transforms velocity to position. Since the disturbances are handled by the inner velocity controller, a proportional controller is used. In this case, all position controllers are cascaded, so the consideration of the bandwidth has to be taken into account in all of them, making each bandwidth three times lower than that of the respective inner loop. Resulting in a bandwidth of 0.23 rad·s^{-1} for the x_I and y_I controllers and 1 rad·s^{-1} for the z_I controller.

V. Results

The controllers have been implemented and tested on the quadcopter. Experimental results are shown for the attitude controller and simulation results are presented for the translational controllers. The latter includes the network and the non-linear system models simulated in MATLAB Simulink.

The obtained model parameters are seen in Table I. The mass and the length have been measured. The coefficients k_th and k_d have been obtained by testing the propellers at different speeds. The moments of inertia have been calculated analytically by considering the quadcopter as a combination of different masses with known moments of inertia.

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	$1.32922 \cdot 10^{-5}$	Ns^{2}rad^{-2}
k_d	$9.39741 \cdot 10^{-7}$	Nms^{2}rad^{-2}
$\overline{\omega}_i$	429	rad s^{-1}

Table I: Parameters used through the analysis and design.

The delay used in the simulation is 31.46 ms and the probability of missed packets is set to zero.

The attitude controller is defined by the designed $\mathbf{Q}$ and $\mathbf{R}$ diagonal matrices, shown below, and the chosen observer poles, which are -11, -12 and -13.

$$\mathbf{Q} = diag\left(\tfrac{1}{0.2^2}, \tfrac{1}{0.2^2}, \tfrac{1}{0.1^2}, \tfrac{1}{0.5^2}, \tfrac{1}{0.5^2}, \tfrac{1}{0.3^2}, \tfrac{1}{0.08^2}, \tfrac{1}{0.08^2}, \tfrac{1}{0.05^2}\right)$$

$$\mathbf{R} = diag\left(\tfrac{1}{25^2}, \tfrac{1}{25^2}, \tfrac{1}{25^2}, \tfrac{1}{25^2}\right).$$

The controllers for $\dot{x}_\mathrm{I}$, $\dot{y}_\mathrm{I}$, $\dot{z}_\mathrm{I}$, x_I, y_I and z_I are

$$C_{\dot{x}_\mathrm{I}} = -C_{\dot{y}_\mathrm{I}} = -0.0038\left(\frac{1+20s}{s}\right) \ ,$$

$$C_{\dot{z}_\mathrm{I}} = -201.8\left(\frac{s+0.8}{s}\right) \ ,$$

$$C_{x_\mathrm{I}} = C_{y_\mathrm{I}} = 0.3 \ ,$$

$$C_{z_\mathrm{I}} = 1 \ .$$

These controllers are discretized using the Tustin method with a sampling time of 35 ms. This is the minimum time in which data can be acquired from the motion tracking system, transmitted to and read by the microcontroller.

Figure 6 shows the attitude controller response when tracking a reference in the roll angle.

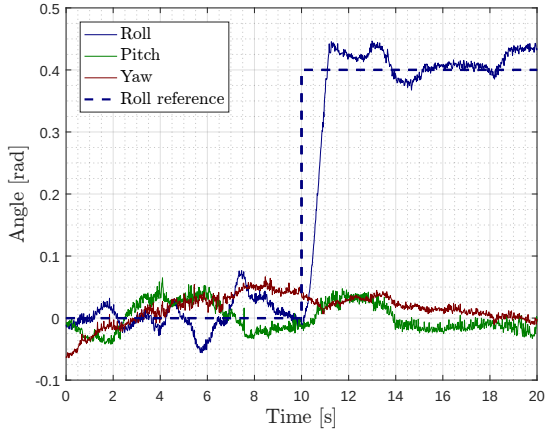

Figure 6: Attitude control response when tracking a reference in roll. The reference for pitch and yaw are kept to zero.

In Figure 7, the simulated step responses of the translational controllers along the three axes are depicted.

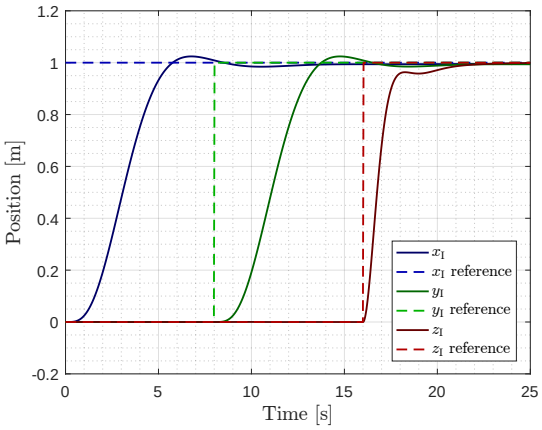

Figure 7: Position control response in the three inertial axes.

VI. Discussion

The results obtained show both the attitude and simulated position response of the quadcopter.

It is seen that the controllers track the references even though the network delay and the sampling rate affects their performance. The network limits the designed bandwidths, especially for the attitude controller. This is due to the limited frequency in which the sensor data is obtained from the motion tracking system. A faster response is achievable if on board sensors for acquiring the attitude are utilized.

Experimental results could not be presented for the translational controllers, as it has not been possible to successfully implement them in due time. The design is however deemed valid, as simulations show that the design should work in reality. The attitude controller is implemented and achieves the given references.

VII. Conclusion

The behavior of a quadcopter has been modeled with a first principles approach. A linear control system has been designed in order to stabilize the quadcopter and control its position.

The control system has been split into attitude and translational controllers. These have been designed considering the main network effects resulting from using an external motion tracking system. The attitude controller has been designed using a state space approach, including state feedback with integral control using LQR and a reduced order observer. The translational control system has been designed by means of classical control methods, which resulted in a cascaded structure including P and PI controllers.

The results show that the design for both the attitude and the translational behavior is able to control the quadcopter in simulation. Moreover, the implementation and test of the attitude controller have been carried out on the quadcopter successfully.

VIII. Future Work

The control system has been designed using an external motion tracking system as sensor for attitude and position. This system limits the operation area of the quadcopter. A future step could be to implement alternative sensors located on the quadcopter to further investigate the capabilities of the control structure and to enable outdoor flight.

IX. Acknowledgments

Henrik Schiøler , associated professor at Aalborg University.

References

[1] *10 incredibly interesting uses for drones*, Web Page, 2014. [Online]. Available: http://dronebuff.com/uses-for-drones/.

[2] Phillip McKerrow, "Modelling the draganflyer four-rotor helicopter", *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 2004.

[3] Daobo Wang Ashfaq Ahmad Mian Mian Ilyas Ahmad, "Backstepping based nonlinear flight control strategy for 6 dof aerial robot", *International Conference on Smart Manufacturing Application*, 2008.

[4] S. McGilvray A. Tayebi, "Attitude stabilization of a four-rotor aerial robot", *43rd IEEE Conference on Decision and Control*, 2004.

[5] Dao-bo Wang Ashfaq Ahmad Mian, "Dynamic modeling and nonlinear control strategy for an underactuated quad rotor rotorcraft", *Journal of Zhejiang University SCIENCE A*, 2008.

[6] B. Daachi A. Mokhtari A. Benallegue, "Robust feedback linearization and ginf controller for a quadrotor unmanned aerial vehicle", *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[7] *Vicon, motion tracking system*. [Online]. Available: www.vicon.com.

[8] Eric W. Weisstein, *Rotation matrix*, MathWorld–A Wolfram Web Resource. [Online]. Available: http://mathworld.wolfram.com/RotationMatrix.html.

[9] Ji Liu Ling Huang Cheng-Chew Lim, "Time delay compensation for positive nonlinear networked control systems with bounded controls", *IEEE International Conference on Fuzzy Systems (FUZZ)*, 2016.

[10] Nikhil Chopra Nirupam Gupta, "Stability analysis of a two-channel feedback networked control system", *Indian Control Conference (ICC)*, 2016.

[11] Dan Henriksson Anton Cervin et al., "How does control timing affect performance? analysis and simulation of timing using jitterbug and truetime", *IEEE Control Systems Magazine*, 2003.

[12] Abbas Emami-Naeini Gene F. Franklin J. David Powell, "Feedback control of dynamic systems", in, 7th Edition. Pearson, 2015, ch. 7, pp. 453–585.