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

Alejandro Alonso García, Amalie V. Petersen, Andrea Victoria Tram Løvemærke,
Niels Skov Vestergaard, Noelia Villarmarzo Arruñada
School of Information and Communication Technology
Control and Automation
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 controller design, which is based upon a model derived by first principle modeling. The control system is divided into attitude and translational. These are designed by 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, keeping the control in a micro processor on the quadcopter. This layout constitutes a distributed system, where network issues, such as delays and missed packets, are taken into account.**

I. Introduction

In the last years, the interest for quadcopters has increased due to the multiple possibilities they offer. Among these, the most well-known ones 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 the three position coordinates and the three orientations. In addition, 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 remote sensing and the attitude controller bandwidth effects on the translational controllers are considered.

In section II, the model of the quadcopter is obtained 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 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 information 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 depicted 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.

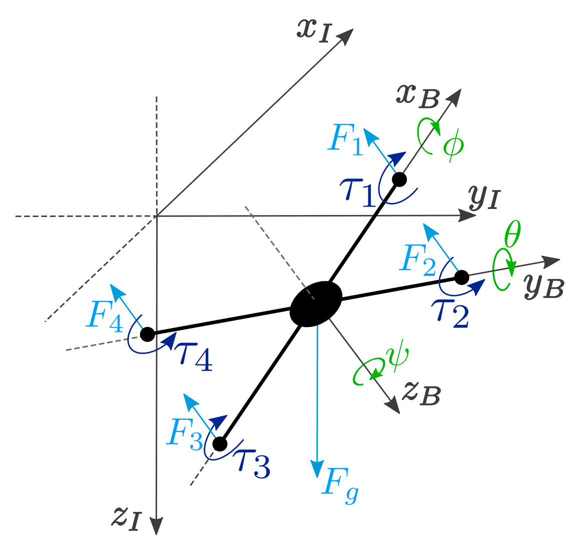

Figure 1: Forces ($\mathbf{F}_i$) and torques ($\boldsymbol{\tau}_i$) acting on the quadcopter, positive references for rotations and positive references for translations in both the inertial and body coordinate frames.

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 characterize its attitude behavior. The positive references for rotational and translational movements are depicted in the figure, as well as the main forces and torques acting on the quadcopter. The positive rotations follow a right-hand rule.

The forces generated in the propeller 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. The first describes the motor and the propeller, the second and the third explain 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 as proportional to the square of the motor rotational speed. The thrust coefficient for each motor is found experimentally. [1]

The rotation also generates a torque on each motor due to the aerodynamic drag. The 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. 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 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.

These equations are used in the attitude and translational models presented below.

B. Attitude Model

The attitude model equations, which are based on Newton's Second Law for rotational movement, state how the thrust forces and the drag torques influence the attitude behavior. They are as follows

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

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. These expressions are

$$\begin{aligned} m\ddot{x}_\mathrm{I} = &- k_\mathrm{th}({\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2 + {\omega_4}^2) \\ &\times (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi), \end{aligned} \tag{6}$$
$$\begin{aligned} m\ddot{y}_\mathrm{I} = &- k_\mathrm{th}({\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2 + {\omega_4}^2) \\ &\times (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi), \end{aligned} \tag{7}$$
$$\begin{aligned} m\ddot{z}_\mathrm{I} = &F_g - k_\mathrm{th}\ ({\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2 + {\omega_4}^2) \\ &\times \cos\phi\cos\theta, \end{aligned} \tag{8}$$

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 angular position of the quadcopter is set to zero in the three angles.

Choosing a zero acceleration linearization point along the z_I axis yields an equilibrium rotational speed. This rotational speed generates a thrust that counteracts the gravitational force. The relation 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

$$m\Delta\ddot{x}_\mathrm{I} = - k_\mathrm{th}({\overline{\omega}_1}^2 + {\overline{\omega}_2}^2 + {\overline{\omega}_3}^2 + {\overline{\omega}_4}^2)\Delta\theta, \tag{13}$$
$$m\Delta\ddot{y}_\mathrm{I} = \quad k_\mathrm{th}({\overline{\omega}_1}^2 + {\overline{\omega}_2}^2 + {\overline{\omega}_3}^2 + {\overline{\omega}_4}^2)\Delta\phi, \tag{14}$$
$$\begin{aligned} m\Delta\ddot{z}_\mathrm{I} = &- 2k_\mathrm{th}\overline{\omega}_1\Delta\omega_1 - 2k_\mathrm{th}\overline{\omega}_2\Delta\omega_2 \\ &- 2k_\mathrm{th}\overline{\omega}_3\Delta\omega_3 - 2k_\mathrm{th}\overline{\omega}_4\Delta\omega_4, \end{aligned} \tag{15}$$

[1] FiXme Note: source on why the thrust and drag torque can be modeled like that.

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, respectively.

III. Network

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

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 TrueTime [11]. It provides the option 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 elapsed until the controller uses the information. It is estimated as the sampling time minus the execution time of the control loop. This yields the maximun delay, thereby considering the worst case scenario.

The missed packets, defined as a constant probability of the controller utilizing 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 manages 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 [2]

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

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

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 be able 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. [12]

Figure 2 shows how these designs are related.

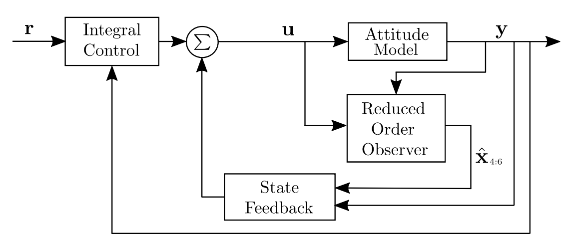

Figure 2: Control structure for the system, including the state feedback with integral action and the reduced order observer.

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

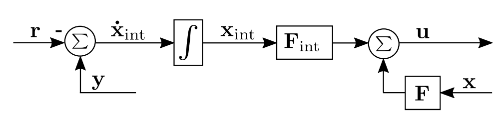

Figure 3: State feedback with integral action in the attitude control structure. $\mathbf{r}$ is the angular reference vector and $\mathbf{x}_{\mathrm{int}}$ is the integral states.

[2]FiXme Note: remove this matrices???

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

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

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

where

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

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

The feedback law is given by

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

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

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

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

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

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

The reduced 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 $\mathbf{L}_{\text{obs}}$ such that the eigenvalues of the matrix $\mathbf{A}_{22}+\mathbf{L}_{\text{obs}}\mathbf{A}_{12}$ 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.

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

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

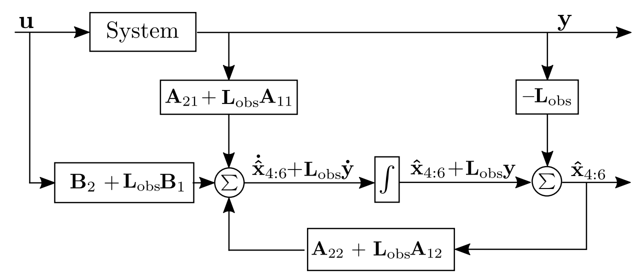

Figure 4: Detailed diagram of the reduced order observer illustrating its implementation.

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.[3]

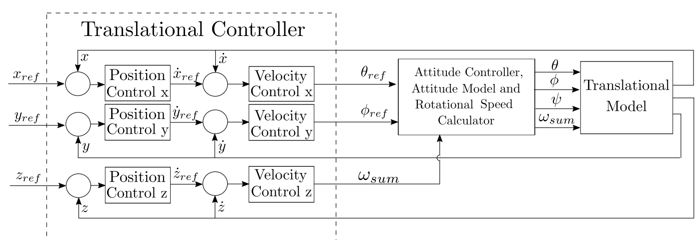

Figure 5: Overview of translational controllers structure.

The x_{I} and y_{I} controllers share similar properties as both their outputs are angle references, θ_{ref} and ϕ_{ref}. This is true as long as the system is close to the linearization point. This implies that ψ_{ref} is set to zero all the time, as it is desired to have zero yaw. This is done, since the translational movement can then be obtained by only rotating the other two axes. The output of the z_{I} controller is the required sum of motor rotational speeds.

[3] FiXme Note: Clarify a bit more the design, use figure 5

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, yielding

$$G_{\dot{x}}(s) = \frac{\dot{x}_\mathrm{I}(s)}{\theta(s)} = \frac{-k_\mathrm{th}(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2)}{ms} \tag{25}$$

$$G_{\dot{y}}(s) = \frac{\dot{y}_\mathrm{I}(s)}{\phi(s)} = \frac{k_\mathrm{th}(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2)}{ms} \tag{26}$$

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

where $G_{\dot{x}}$, $G_{\dot{y}}$ and $G_{\dot{x}}$ are the plants used to design the velocity controllers in x_I, y_I and z_I directions respectively, $\dot{z}_\mathrm{I}$ is the velocity in the z_I direction, ω_sum is the sum of the rotational speeds of the motors and $\overline{\omega}_\mathrm{sum}$ is the sum of the rotational speeds in equilibrium.

The three plants contain an integrator that can handle steady state errors and output disturbance. However, is there are input disturbances, which can affect the rotational speeds of the motors or external causes such as wind, an integrator term in the controller is needed for eliminating the error that they cause. Finally, a zero is then added to remove the marginal instability due to the presence of only two integrators in the system and the gain is adjusted to reduce the oscillating behavior. It is worth mentioned that, since the plants for the x_I and z_I velocities have a negative gain, the controller needs to compensate it with a negative gain as well.

Since the controllers for $\dot{x}_\mathrm{I}$, $\dot{y}_\mathrm{I}$ use the attitude controller as an inner loop to obtained the required angles, a special attention should be put on the bandwidth of the velocity controllers. If it is too fast, the inner one [4]. Based upon a rule of thumb, which suggest a three to five times slower outer loop, the gain is designed such that the system has a bandwidth that is three times lower than the attitude control loop, which is 2 rad s^{-1}, to reduce the effect of its dynamic in the velocity controllers. This yields a bandwidth of around 0.7 rad s^{-1} for the velocity controllers in x_I and y_I directions.

The plants of the outer loops (position control loops) contain only integrator that transforms velocity to position. Since the disturbances are handled by the inner velocity controller, a proportional controller is used. In this case, there exists an inner loop in the three axes, so the consideration of the bandwidth has to be taken into count in all of them. For x_I and y_I the final bandwidth is 0.23 rad s^{-1}, while for z_I, is three times lower than the inner closed loop for $\dot{z}_\mathrm{I}$ yielding 1 rad s^{-1}.

V. Results

The controllers have been implemented and tested in the real quadcopter. The real results for the attitude controller and simulation results with the network and the model simulated in MATLAB Simulink for the translational controllers are presented.

The obtained model parameters can be seen in Table I.

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 though the analysis and design.

The mass and the length have been measured, k_th and k_d have been obtained through testing the propellers when rotating at different speeds and the moments of inertia have been calculated analytically considering the quadcopter as a combination of different masses with known moment of inertia.

The value of delay used in the simulation [5] milliseconds and the packet loss probability 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 $[-11, -12, -13]$.

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

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

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

$$C_{\dot{x}_\mathrm{I}} = -C_{\dot{y}_\mathrm{I}} = -0.0038\frac{1+20s}{s}, \quad C_{\dot{z}_\mathrm{I}} = -201.8\frac{s+0.8}{s},$$

$$C_{x_\mathrm{I}} = C_{y_\mathrm{I}} = 0.3, \qquad C_{z_\mathrm{I}} = 1 .$$

These controllers are discretized using the Tustin method with a sampling frequency of 28 Hz. This is the fastest available frequency in which data can be acquired from the motion tracking system, transmitted to the quadcopter and read by the microcontroller.

?? and 6 show the attitude controller response when tracking a reference in pitch angle.

[4] FiXme Note: insert source, Noelia, dropbox

[5] FiXme Note: WRITE NUMBER

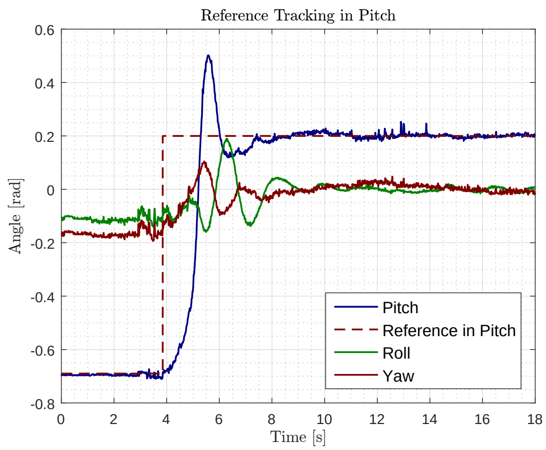

Figure 6: Real attitude control response when tracking a reference in pitch.

In **??**, the simulated step responses of the translational controllers along the three axes are depicted.

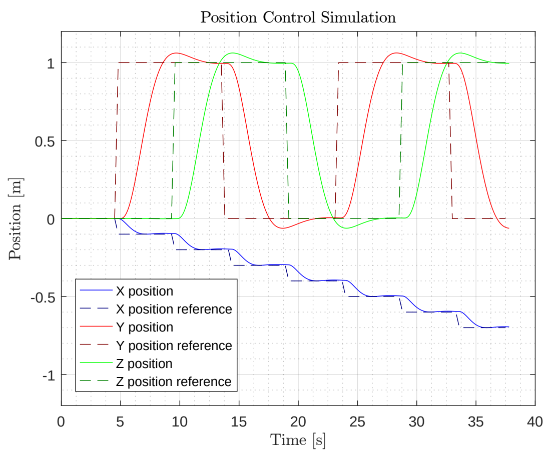

Figure 7: Position control results in the three inertial axes directions. The references given to the control system are shown with dashed lines.

VI. Discussion

[6] The results obtained in the simulations show both the attitude and position response of the quadcopter.

It is seen that the controllers achieve the desired references even though the network delay and the sampling rate affects the performance by reducing stability. This effects limit the design bandwidth of the controllers. This occurs due to the limited frequency in which the sensor data is obtained from the motion tracking system through the wireless connection.

It is also worth observing how the attitude controller shows a permanent error with respect to the reference. This is generated as a result of the integral controller design because it assumes a constant reference applied to it. This issue, though, does not affect the final position of the quadcopter.

[6]FiXme Note: why the controllers are slow?, the delay is high

VII. Conclusion

The behavior of a quadcopter has been modeled by first principles of physics. A linear 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 former 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. The results obtained from the design show that both the attitude and the translational behavior of the quadcopter has been successfully controlled.

VIII. Future Work

[7] The control system has been designed using a motion tracking system as an attitude sensor. In order to improve the result while keeping the same control structure, an inertial measurement unit could be installed on the quadcopter.

References

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

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

[3] D. W. Ashfaq Ahmad Mian Mian Ilyas Ahmad, "Back-stepping based nonlinear flight control strategy for 6 dof aerial robot", *International Conference on Smart Manufacturing Application*, 2008.

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

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

[6] B. D. 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] E. W. Weisstein. (), [Online]. Available: http : / / mathworld.wolfram.com/RotationMatrix.html.

[9] J. L. 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.

[7]FiXme Note: Should we talk about non linear control?

[10] N. C. Nirupam Gupta, “Stability analysis of a two-channel feedback networked control system”, *Indian Control Conference (ICC)*, 2016.

[11] D. H. 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] A. E.-N. Gene F. Franklin J. David Powell, “Feedback control of dynamic systems”, in, 7th Edition. Pearson, 2015, ch. 7, pp. 453–585.