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 controller design, which is based upon a model that is derived by first principle modeling. This is later linearized since it is desired to use linear controllers. The system is divided into an attitude and translational control loops. 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 packet losses, are 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 six degrees of freedom, the three position coordinates and the three orientations, and there are only four actuation variables, namely the motor rotational speeds. **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. Another 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 Wang 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 linearization and a modified optimization control method.

This paper addresses the performance achievable using the chosen linear attitude control strategy, the influence of remote sensing on the attitude control, the influence of the attitude control bandwidth on the translational controllers and the performance of a cascaded control structure with classical linear translational controllers.

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. With the linearized equations, controllers for attitude and translational behaviors are designed in section IV. The attitude controller is obtained by means of a state space representation while the translational controller is designed using classical control. 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 affects the control loop is also presented, this is done in section III. In section V, the simulations and experimental results of the designed controllers are presented. They are after 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.

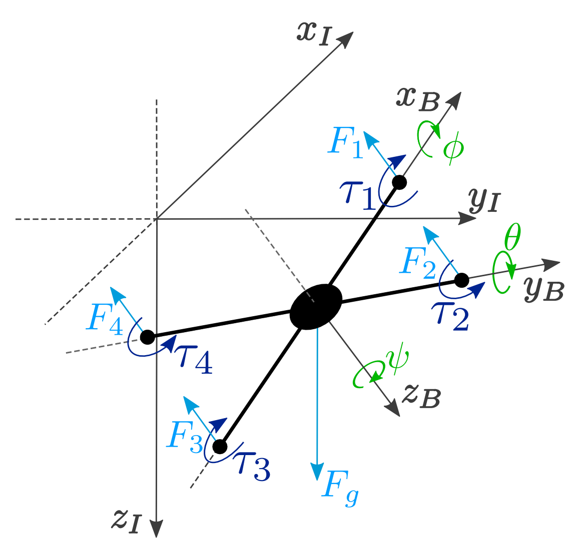

Figure 1: Forces ($\mathbf{F}_i$) and torques ($\boldsymbol{\tau}_i$) acting on the quadcopter and the positive references chosen for rotations and translations in both inertial and body coordinate frames.

As it is 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 positive rotations follow a right-hand fashion.

The forces generated in the propeller are readily 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 **rotationmatrix** This means that any rotation is described as three rotations around the x_{B} axis first, 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 presents the attitude response of the quadcopter and the third explains how the translational variables of the system evolve.

[1]

A. Motor and Propeller

The four motors in the quadcopter generate a rotation in the propellers that creates the force that lifts the quadcopter. The thrust force can be modeled as proportional to the square of the motor rotational speed. The thrust coefficient for each motor is found experimentally. [2]

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

[1] FiXme Note: In the modeling section, what are your assumptions? What is the reason for not including model parts such as Gyroscopic torque, Air friction/drag on the quadcopter itself (even though you include it on the propellers), Aerodynamic lift, Mass placement and separation of masses, Coriolis acceleration?

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

B. Attitude Model

The attitude model equations, which are based on Newton's Second Law for rotational movement, 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, 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 parameters of the model are found in section V.

The expressions above state how the thrust forces and the drag torques generated on the propellers affect the attitude behavior of the quadcopter.

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 that act on the system are those from the propellers and the gravitational force. These expressions are

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

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 respectively and F_g is the gravitational force acting on the quadcopter. The parameters of the model are found in section V.

It is worth mentioning that, as the thrust forces always point in the negative z_{B} direction, the accelerations along x_{I} and y_{I} directions are zero when pitch and roll angles are zero.

D. Linearization

The model equations are linearized using the first order Taylor approximation around an equilibrium point of the system. This point is the hovering position, which implies that the attitude and translational accelerations and velocities are zero. The angular position of the quadcopter is also set to zero in the three angles.

Choosing a zero acceleration linearization point along the z_{I} axis yields an equilibrium rotational speed so that the necessary thrust is generated to compensate for the gravitational force. The relation is expressed as

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

The resulting equations for the attitude model after the linearization are

$$J_x\Delta\ddot{\phi} = 2k_{\text{th}}L\overline{\omega}_4\Delta\omega_4 - 2\ k_{\text{th}}L\overline{\omega}_2\Delta\omega_2 \tag{10}$$
$$J_y\Delta\ddot{\theta} = 2k_{\text{th}}L\overline{\omega}_1\Delta\omega_1 - 2k_{\text{th}}L\overline{\omega}_3\Delta\omega_3 \tag{11}$$
$$J_z\Delta\ddot{\psi} = 2k_{\text{d}}\overline{\omega}_1\Delta\omega_1 - 2k_{\text{d}}\overline{\omega}_2\Delta\omega_2 + \ 2\ k_{\text{d}}\overline{\omega}_3\Delta\omega_3 - 2\ k_{\text{d}}\overline{\omega}_4\Delta\omega_4 \tag{12}$$

where $\Delta\ddot{\phi}$, $\Delta\ddot{\theta}$ and $\Delta\ddot{\psi}$ are the changes in rotational acceleration from 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

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

where $\Delta\ddot{x}_{\text{I}}$, $\Delta\ddot{y}_{\text{I}}$ and $\Delta\ddot{z}_{\text{I}}$ 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 modeled by considering two effects: the delay and the loss of packets in the communication channel.

The theoretical modeling of these influences has been studied by several researchers in order to obtain a criteria for finding maximum allowable delay and packet loss for the control system to remain stable **ling nirupam** However, these approaches often lead to an increased complexity as the network effect is included in the model of the system.

Stability of the system when influenced by the wireless network is instead analyzed using the TrueTime Simulator **TrueTimeNew** 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, ensure 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 and it is estimated as the sampling time minus the execution time of the control loop. With this delay model, the worst case scenario is considered.

The packet loss, defined as a constant probability of loosing a packet, is found experimentally by sending a large amount of packets and examining how many of these are available for the controller. In the experiment, it has been found that more packets are received than control loops are executed, that is, the most recent packet is always available for the controller and, therefore, the packet loss probability is zero.

IV. Control

The control of the system is divided into two control systems. One handles the attitude and the other controls the translational behavior of the quadcopter. These two are related such that the translational controller sets the references for the angles handled by the inner controller. When designing the controllers, the network is considered to account for the delays the distributed system introduces.

A. Attitude Controller

[3] The attitude controller for the quadcopter 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 of the attitude system 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 [4]

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

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

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

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

[3] FiXme Note: Explain more on why this design process is carried out
[4] FiXme Note: remove this matrices???

velocities of the quadcopter. Due to the separation principle, both subsystems can be designed independently. **ssReference**

Figure 2 shows how these designs are related.

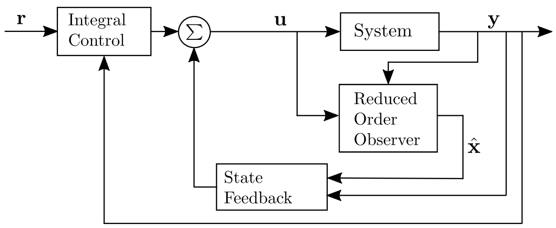

Figure 2: Control structure for the system, including the state feedback, the integral controller and the reduced order observer.

The design of the state feedback and integral control is shown in Figure 3.

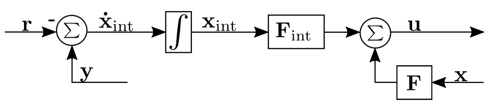

Figure 3: State feedback and integral controller in the attitude control structure.[5]

Three states, $\mathbf{x}_{\text{int}}(t)$, are added to the already existing state vector to track the two references for ϕ and θ angles and reject input disturbances in the three angles. This leads to the extended system

$$\dot{\mathbf{x}}_{\text{e}}(t) = \mathbf{A}_{\text{e}}\mathbf{x}_{\text{e}}(t) + \mathbf{B}_{\text{e}}\mathbf{u}(t) + \begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix} \mathbf{r}(t) \qquad (16)$$

$$\mathbf{y}(t) = \mathbf{C}_{\text{e}}\mathbf{x}_{\text{e}} \qquad (17)$$

where

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

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

The feedback law for this design approach is given by

$$\mathbf{u}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{F}_{\text{int}}\mathbf{x}_{\text{int}}(t) \ . \qquad (18)$$

The feedback matrix, $\mathbf{F_e}$ is obtained 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 required. LQR is based on minimizing the cost function

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

In this equation, the matrices $\mathbf{Q}$ and $\mathbf{R}$ must be selected as positive semi-definite and positive definite matrices that penalize, in the cost function, the error of the different states and inputs of the system.

The selection of these matrices is done using Bryson's rule, which defines $\mathbf{Q}$ and $\mathbf{R}$ as diagonal matrices whose elements are

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

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

The values are selected so the controller is stable and the states converge to the equilibrium point and the references are tracked as fast as possible. This gain selection process is done using the simulation of the model, which also considers the network effects such as delay and packet loss. The values of $\mathbf{Q}$ and $\mathbf{R}$ are then used to find the optimal feedback matrices that minimize the cost function (19) as

$$\mathbf{F}_{\text{e}} = -\mathbf{R}^{-1}\mathbf{B}_{\text{e}}{}^T\mathbf{P} \ . \qquad (22)$$

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

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

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.

Since certain states in the system are not measured, an observer estimates the remaining states 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 $\mathbf{A}_{22} + \mathbf{L}_{\text{obs}}\mathbf{A}_{12}$ give the desired dynamics for the estimation **ssReference** This is done splitting the original system matrices into the submatrices.

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

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.

$$\dot{\hat{\mathbf{x}}}_{4:6} = \mathbf{A}_{21}\mathbf{y} + \mathbf{A}_{22}\hat{\mathbf{x}}_{4:6} + \mathbf{B_2}\mathbf{u} + \mathbf{L}_{\text{obs}}(\mathbf{A}_{12}\hat{\mathbf{x}}_{4:6} - \mathbf{A}_{21}\mathbf{x}_{4:6}) \qquad (24)$$

[7] This estimation of $\mathbf{x}_{4:6}$ is implemented as shown in Figure 4.

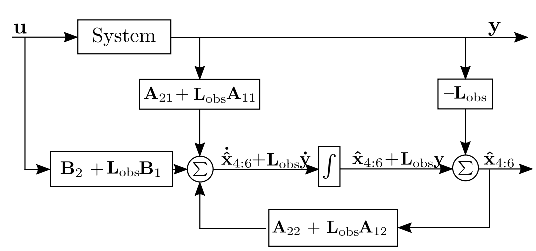

Figure 4: Detailed diagram of the reduced order observer that shows its implementation.

[6]FiXme Note: Formulate desired dynamics in another

[7]FiXme Note: put proper equation and think if the picture is needed

B. Translational Controller

The translational controllers are structured 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 5.[8]

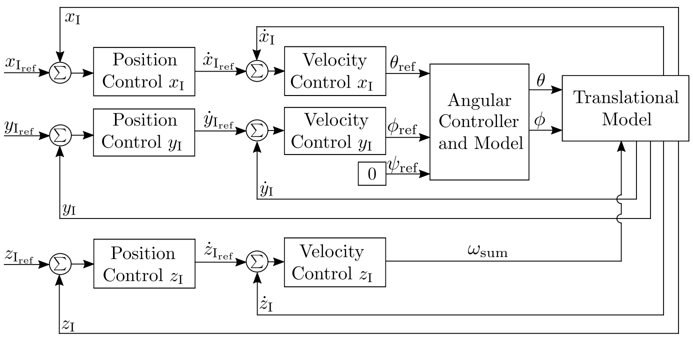

Figure 5: Overview of translational controllers structure.

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.

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 [9]. Based upon a rule of thumb, which suggest a 3-5 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 transfer functions for the translational position controllers, three times less bandwidth than the inner velocity loop. The plant of the outer loop is considered only as an integrator that transforms velocity to position. As for the inner loop, the controller of the outer loop is a proportional controller.

V. Results

The controllers have been implemented and tested in the real quadcopter. The obtained results are also compared with the network and the model have been simulated in MATLAB Simulink in order to generate results presented in this section.

The 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}	$1.32922 \cdot 10^{-5}$	$\mathrm{Ns^2rad^{-2}}$
k_{d}	$9.39741 \cdot 10^{-7}$	$\mathrm{Nms^2rad^{-2}}$
$\overline{\omega}_i$	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 [10] 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 $[-11, -12, -13]$.

$\mathbf{Q} = diag\left(\frac{1}{111^2}, \frac{1}{222^2}, \frac{1}{333^2}, \frac{1}{444^2}, \frac{1}{555^2}, \frac{1}{666^2}, \frac{1}{777^2}, \frac{1}{999^2}, \frac{1}{888^2}\right)$
$\mathbf{R} = diag\left(\frac{1}{111^2}, \frac{1}{222^2}, \frac{1}{333^2}, \frac{1}{444^2}\right)$

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

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

[10] FiXme Note: WRITE NUMBER

[11] The translation velocity controllers for x, y and z are

$$C_{\dot{x}_{\mathrm{I}}} = -C_{\dot{y}_{\mathrm{I}}} = TOWRITE \qquad C_{\dot{z}_{\mathrm{I}}} = -201.8\frac{s+0.8}{s} ,$$

while the position controllers are

$$C_{x_{\mathrm{I}}} = -C_{y_{\mathrm{I}}} = TOWRITE \qquad C_{z_{\mathrm{I}}} = -201.8\frac{s+0.8}{s} .$$

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 and transmitted to the quadcopter.

The simulation results obtained with these parameters and controllers are shown in Figure 6.

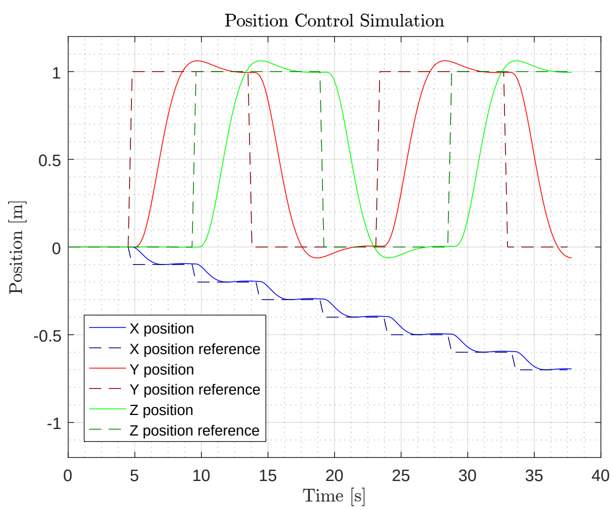

Figure 6: Position control results in the three inertial axes directions. The references given to the control system are shown with dashed lines.

The inner attitude controller results are also included and shown in Figure 7 so the performance of the attitude control can be evaluated.

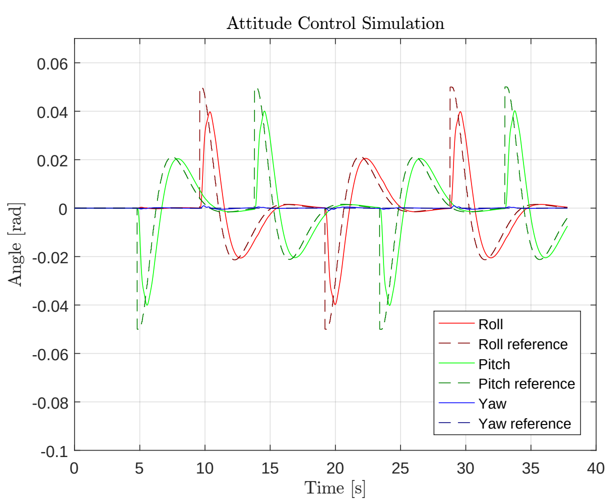

Figure 7: Attitude control results in the three angles. The references given to the attitude control system are shown with dashed lines.

[11] FiXme Note: check values in matrices

VI. Discussion

[12] 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 reference 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 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.

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

[13] 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.

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

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