Università degli studi di Roma *La Sapienza*

Dipartimento di Ingegneria Informatica, Automatice e Gestionale Antonio Ruberti

Modeling and Simulation of different type of Rotary-Wing Vertical Take-Off and Landing vehicles (*RWVTOLs*)

Elective course in Robotics
Module : *Modelling and control of Multi Rotor UAV*
Prof. Marilena Vendittelli

Marco Capotondi

Indice

Capitolo 1

Glossary

Here are shown definitions of terms and acronyms that will be used in the report.

- **RWVTOL**, rotary wing based vehicle that can take off and land in vertical posture
- **VTOL**, vertical take off and landing vehicle
- **CoM**, center of mass of a body
- **Hexacopter**, rotor wing based drone with six rotors
- **Quadrotor**, rotor wing based drone with four rotors
- **MEMS**, that stand for *Microelectromechanical systems*, is the technology of microscopic devices, particularly those with moving parts
- **Stationarity**, states or conditions that don't change in time
- **Linearization**, approximation of non linear system in a linear one given particular conditions
- **Newton-Euler equations**, in classical mechanics these equations describe the combined translational and rotational dynamics of a rigid body
- **Open Loop Simulations**, simulations of dynamical systems considering only the process with arbitrary inputs, without considering output of the system
- **Closed Loop Simulations**, simulations of dynamical systems that modify input considering the output of the same system
- **Runge Kutta integration**, family of implicit and explicit iterative methods used in temporal discretization for the approximate solutions of ordinary differential equations
- **Regulation** stands in robotics the task of driving the state of a dynamical system from an initial value to a reference fixed state (as example fixed positions and 0 velocities)
- **Trajectory Tracking** stands in robotics the task of driving the state of a dynamical system from an initial value to a reference state that changes in time (as trajectories for mechanical systems)

Capitolo 2

Introduction

In this project will be analyzed different types of *RWVTOL*,. In particular the focus of the project is the modeling and simulations of three type of vehicles:

- Single rotor helicopter model
- Hexacopter
- Quadrotor with tilting propellers

2.1 Overview on RWVTOL

Is defined VTOL a vehicle able to take off and land in a vertical posture. An example of this type of vehicle is Helicopter, while as counter example can be mentioned classical airplane.
In particular is defined *RWVTOL* a VTOL that uses main rotor(s) to generate lift in one privileged direction allowing to counterbalance its weight and aerodynamic perturbations. Also in this case can be mentioned as example helicopters, but also Quadrotors, Hexacopters. The technology development (eletrical drives very efficient, computational power, camera systems, improvements on batteries, MEMS tech on sensors) allowed introduction of automated control system on these vehicles (as well is happening in automotive).
The main advantage of VTOL vehicles is the possibility of the taking off without the need of a runway, so the portability of the vehicle in environment smaller. In general, their versatile flight capability have found applications in different scenarios (civil and military environment).

The main problems of these systems are :

- Generation and control of thrust and torques in order to achieve goals as trajectory tracking or cost functions optimization
- Balance parasite torques that comes out from aerodynamic effects
- Reject external and internal disturbances on the system

Capitolo 3

Modeling and Open Loop Simulations

Mechanical modeling of this type of vehicles can be very complex, considering structure and aeredinamic analysis. Despite of, purpose of the project is not to model parameters that derives from deep mechanical analysis, but schematize dynamics of the system in order to have a model that can be eventually used for controls and vision systems.
However these models are simplified, **linearized** around particular conditions and **stationarity conditions** assumed.

3.1 Representation of Rigid Body in 3D

Is defined rigid body a solid body in which deformation is zero or so small that can be neglected. The consequences are that distance between any two given points on a rigid body remains costant in time regardless of external forces.
Pose of rigid body is represented by:

- Linear position of the body, represented by the relative position of a reference frame centered in the CoM of the rigid body respect to an inertial reference frame chosen

- Posture of the rigid body, represented by the orientation of the reference frame centered in the CoM of the rigid body and with same orientation respect to an inertial reference frame chosen

Representation of rigid body has to care around the presence of two reference frame, the non inertial one centered in the body and the inertial one, external to the body. Usually dynamics of the system is evolved in the reference frame of the body and then represented in the inertial reference frame.

3.1.1 Posture Representation

Posture of the rigid body is represented basically from 3D rotation matrix between reference frame of the body RF^B and inertial reference frame RF^i. However is known, from *Group Theory*, that 3D rotation matrix forms a group, denoted as **SO(3)**, group of orthonormal matrix with determinant equal to 1. Are also well known the possible parameterization of elements in that group:

- *Euler Angles* (ϕ, θ, ψ) representing a product of rotations about (X, Y, Z) mobile axis
- *Unitary Quaternion*, an extension of complex number system, in the form $q = q_0 + iq_1 + jq_2 + kq_3$, where q_0, q_1, q_2, q_3 are real numbers and i, j, k are quaternion units.
- *Axis angles* representation ($\mathbf{r}$,θ), in which the unit vector $\mathbf{r}$ represents the direction of the axis of rotation while θ describes the magnitude of the rotation about that axis.

All of these parametrization are local charts, so they have mutiple-values (lack of injectivity) and singularities. In particular, Euler Angles representation has the problem of **Gimbal Lock**, while quaternion representation has a **Double Cover**, since $\vec{q}$ and $\vec{-q}$ gives the same rotation.
For the open loop models of Helicopter and Quadrotor were used posture representation with rotation matrices, while for the Hexacopter was used quaternion representation. Despite of, for the closed loop simulations of Hexacopter and Helicopter were used Euler Angles representation, while for Quadrotor still used the rotation matrices.

3.1.2 Dynamics of Rigid Body in 3D

Generally, all models developed for description of rigid body's dynamics are based on the *Newton Euler* equations and posture's representation with rotation matrices.
With respect to a coordinate frame whose origin coincides with body's center of mass (RF^B) , forces and torques are described by following equations :

$$\begin{pmatrix} \mathbf{F}_B \\ \boldsymbol{\tau}_B \end{pmatrix} = \begin{pmatrix} m\mathbf{I}_3 & 0 \\ 0 & \mathbf{I}_{\mathrm{cm}} \end{pmatrix} \begin{pmatrix} \mathbf{a}_{\mathrm{cm}} \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} 0 \\ \boldsymbol{\omega} \times \mathbf{J}_{\mathrm{cm}}\, \boldsymbol{\omega} \end{pmatrix},$$

- $\vec{F}_B$ = total force acting on the center of mass
- m = mass of the body
- I_3 = the 3x3 identity matrix
- $\vec{a}_{cm}$ = acceleration of the center of mass
- $\vec{v}_{cm}$ = velocity of the center of mass
- $\vec{\tau}_B$ = total torque acting about the center of mass
- J_{cm} = moment of inertia about the center of mass
- $\vec{\omega}$ = angular velocity of the body
- $\vec{\alpha}$ = angular acceleration of the body

Thus, these equations are calculated in reference frame centered in the *CoM* of the rigid body. So both forces and torques need to be represented in the inertial frame RF^i. The general way to represent vehicle's equations of motion in the inertial frame reference is the following.

Consider a vehicle moving in the 3D-space and subjected to a force vector $F_B \in R^3$ and a torque vector τ_B, applied at its center of mass (CoM), expressed both of them in the body's frame. Then let's define $\vec{v} \in R^3$ vehicle's linear velocity vector and $\vec{\omega} \in R^3$ body angular velocity vector, both of them expressed in the inertial reference frame RF^i. Then $\vec{x} \in R^3$ position of the CoM in RF^i and R the rotation matrix of the body relative of the body frame relative to inertial frame and $S(\omega)$ skew-symmetric matrix of ω.

$$\begin{pmatrix} \dot{x} \\ \\ m\dot{v} \\ \\ \dot{R} \\ \\ J\dot{\omega} \end{pmatrix} = \begin{pmatrix} Rv \\ \\ -mS(\omega)v + F_B \\ \\ RS(\omega) \\ \\ -S(\omega)J\omega + \tau_B \end{pmatrix}$$

Usually in *UAV* τ_B contains torque generated from **propellers**, torque generated from **external forces** (disturbances as wind), **gyroscopic** torque and ofter **coupling torque** between resultant Thrust and Momenta.

3.2 Helicopter Model

The first model analyzed is the classical single rotor helicopter model.

Below is shown a picture that shows geometrical parameters of the helicopter.

Let's define:

- $[l_m, y_m, h_m]^T$ and $[l_t, y_t, h_t]^T$ coordinates of the center of the main and tail rotors, expressed in RF^B
- g the magnitude of the gravity
- $e_3 = [0, 0, 1]^T$

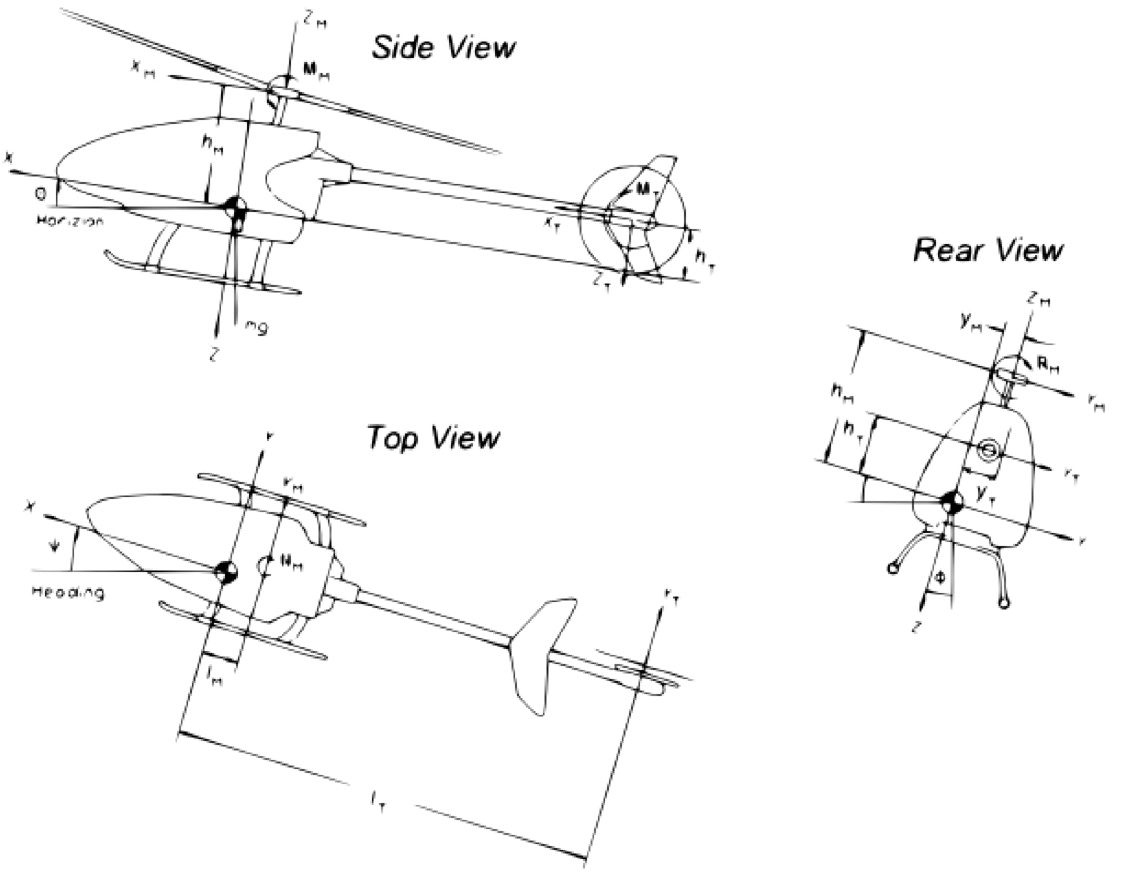


Figura 3.1: Helicopter's model representation

- a and b respectively the longitudinal and lateral tilts of the path plane of the main rotor about the shaft.

The model is based on few approximations, often used for control design:

- hypothesis of $[T_M, b, a, T_t]^T$ vector as available control input
- a,$b \ll 1$
- neglect *rolling moment stiffness* and *pitching moment stiffness* and parasites torques and forces
- $h_M = h_t = h$ and $y_M \ll 1$

With these approximations is obtained the following model :

$$\begin{pmatrix} F_B \\ \\ \tau_B \end{pmatrix} = \begin{pmatrix} -Te_3 + \Sigma_R \Gamma + R^T mge_3 + F_{ae}^B \\ \\ \Gamma + T\Sigma_T + \Gamma_{ae}^B \end{pmatrix}$$

In which :

- $\Gamma = \begin{pmatrix} -h & 0 & h \\ 0 & -h & 0 \\ l_m & 0 & -l_t \end{pmatrix} \begin{pmatrix} bT \\ aT \\ T_t \end{pmatrix}$

- $\Sigma_R = \begin{pmatrix} 0 & \frac{1}{h} & 0 \\ -\frac{1}{h} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = -\frac{1}{h}S(e_3)$

- $\Sigma_T = \begin{pmatrix} -y_M \\ l_M \\ 0 \end{pmatrix}$

3.2.1 Open Loop Numerical Simulation Methodology

In absence of aerodynamics drag coefficients numerical values, F_{ae}^B and Γ_{ae}^B was posed to 0.
The numerical simulation is essentially divided in the following steps:

- F_B and τ_B calculation
- $\alpha = \frac{d}{dt}(\omega)$ calculation and numerical integration (all numerical integration have been realized using **ode45** function of Matlab, so *Runge-Kutta* fourth order integration) in order to calculate ω
- From ω calculation of the derivative of the rotation matrix $\frac{d}{dt}(R) = S(\omega)R$
- Then from $\dot{R}$ numerical integration component by component and extraction of the rotation matrix (this part will be analyzed in a specific section).
- Calculation of linear acceleration in reference frame of the body $\vec{a}_B$
- Conversion of $\vec{a}_B$ from RF^B to RF^i, $\vec{a}_i = R_B^i \vec{a}_B$
- Loop closure with implicit delay for variables updating in the next steps (as ω, R,etc...)

3.2.2 Numerical Integration of Rotation Matrix

In all these kind of simulations, the most critical part is usually the numerical integration of the posture dynamics of the system. The posture representation used in this model is the rotation matrix representation. Is well known that in a rigid body time derivative of the rotation matrix is in linear relation with the matrix itself : $\dot{R} = S(\omega)R$
This equation can be used to integrate numerically R. Despite of, the main problem of this approach is the the numerical drift of the integration , that can obtain a matrix that is not a pure rotation matrix (**Orthonormality**, unitary determinant) but an affine matrix (a generic rototranslation). The extraction of the closest rotation matrix from this affine transformation is called the **Orthonormal Procrustes problem**.
Formally :

$R = argmin_{\Omega} ||\Omega A - B||_{Frobenious}$ s.b.t

$\Omega^T \Omega = I$

$\Omega_i^T \Omega_j = \delta_{i,j}$

That is equivalent to described problem considering decomposition of the matrix affine result of the numerical integration as $M = BA^T$. Given the SVD decomposition of the matrix $M = U\Sigma V^T$, the solution of the optimization problem is the matrix $R = U\Sigma' V^T$, where Σ' is diagonal matrix with the smallest singular value replaced by $sign(det(UV^T)$ and other singular values replaced with $+1$ (these substitutions in order to keep determinant to $+1$).

3.3 Hexacopter model

The second model analyzed is the hexacopter model. Hexacopter, differently from classic helicopter, is usually a vehicle with geometric parameters so small that can be defined as *micropters*. This kind of RWVTOL

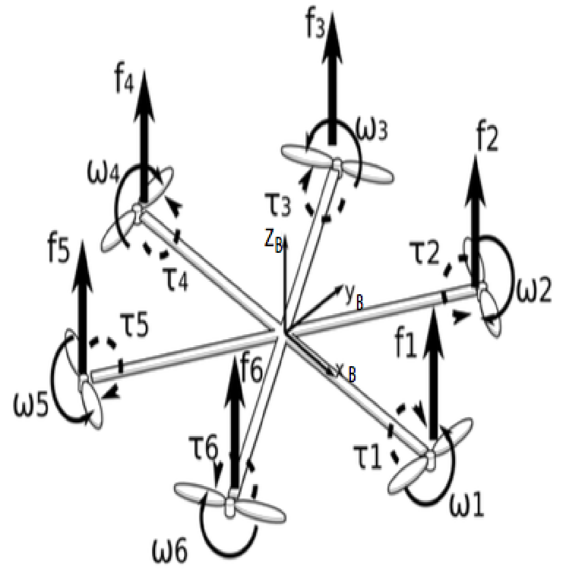


Figura 3.2: Hexacopter's model representation

consists in six rotors, with three pairs of counter rotation fixed pitch blades. The drone is controlled actuating angular velocities of the rotors which are spun by electric motors.
Below is shown a picture of the vehicle.

Let's define an inertial reference frame RF^i in which absolute linear position $(X, Y, Z)^T$ of the hexacopter is defined. Then is defined the body fixed frame RF^B, centered in the hexacopter center of gravity and oriented as shown in the picture. In order to define position of the drone, representation used in the inertial frame RF^i is :

- $\vec{\xi} = (x, y, z)^T$
- $\vec{q} = (q_0, q_1, q_2, q_3)^T$

while in the body frame RF^B is :

- $\vec{\xi}_B = (x_B, y_B, z_B)^T$
- $\vec{\eta} = (\phi, \theta, \psi)^T$, Euler angles representation that won't be used because angular velocity RF_B ν will be represented in RF^i

A quaternion is generally represented in the form $a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ where a, b, c, d are real numbers and i, j, k are fundamental *quaternion units*.
The transformation of the linear coordinates from RF^B to RF^i can be expressed by $\xi = Q\xi_B$, with Q the following matrix

$$\mathbf{Q} = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}$$

while, considering velocities, the conversion between angular velocity from RF^B to RF^i is $\dot{q} = S\nu$, with

- $\nu = (p, q, r)^T$ angular velocity in RF^B

- $S = \frac{1}{2}\begin{pmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix})$

3.3.1 Dynamics equations of the hexacopter

The basic step is schematization of the force generated by the single rotor, represented in the reference frame of the body, $f_i = (0, 0, f_i)^T$. Most simple relation between angular velocity of the single rotor and its force generated is $f_i = k\omega_i^2$, with k lift costant.
From that is possible to obtained total thrust T_B as $T_B = (0, 0, T)^T$, with T= $k\sum_{k=1}^{6} w_i^2$. All of these quantities are expressed in RF^B.
Then the final differential equation, for inertial positional coordinates ξ, is

m$\ddot{\xi} = F_g + QT_B$

Moreover, let I be the inertia matrix of the hexa. The drone has a symmetric structure with respect to (X_B, Y_B, Z_B), having so $I = diagonal(I_{xx}, I_{yy}, I_{zz})$.
Angular acceleration and velocity of the i-th rotor create a torque $\tau_M = b\omega_i^2 + I_{Mi}\dot{\omega}_i$, with b drag constant and I_{Mi} inertia matrix of the rotor.
In order to relate these torques with rotational dynamics of the drone, according to geometry of the system, is possible to calculate total moment on Euler's angles:

$\tau_\phi = \frac{3}{4}kl(\omega_2^2 + \omega_3^2 - \omega_5^2 - \omega_6^2)$

$\tau_\theta = (-\omega_1^2 - \frac{\omega_2^2}{4} + \frac{\omega_3^2}{4} + \omega_4^2 + \frac{\omega_5^2}{4} - \frac{\omega_6^2}{4})$

$\tau_\psi = b(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2 - \omega_5^2 + \omega_6^2) + I_M(\dot{\omega}_1 + \dot{\omega}_2 + \dot{\omega}_3 + \dot{\omega}_4 + \dot{\omega}_5 + \dot{\omega}_6)$

Inverting equation of rotational dynamics

$I\dot{\nu} + \nu \wedge (I\nu) + \Gamma = \tau_B$

$\dot{\nu} = I^{-1}(-\nu \wedge (I\nu) - \Gamma + \tau_B)$

with $\Gamma = I_r(\nu \wedge (0, 0, w_\Gamma)^T)$

So final equations used for simulation are :

m$\ddot{\xi} = F_g + QT_B$

$\dot{\nu} = I^{-1}(-\nu \wedge (I\nu) - \Gamma + \tau_B)$

$\dot{q} = S\nu$

3.3.2 Open Loop Numerical Simulation Methodology

The numerical simulation is essentially divided in the following steps:

- Calculation of moment on *Euler Angles* $(\tau_\phi, \tau_\theta, \tau_\psi)^T$ and Thrust T from angular velocity of rotors (input)

- Then integration of rotational dynamics, calculating $\dot{\nu} = (\dot{p}, \dot{q}, \dot{r})^T$ (angular acceleration in RF^B), integrating it and calculate $\dot{q} = S\nu$, and from that integrate again to calculate $\vec{q}$, representation of the posture.

- After that new posture has been calculated, are calculated acceleration in the inertial frame $\ddot{\xi} = F_g + QT_B$, integrated in order to have 3D position at that time step.

3.3.3 Integration of quaternions

Numerical integration of quaternions, differently from rotation matrix, needs only renormalization of the quaternion, that can lost his property of unitary norm.

$|\vec{q}| = \sqrt{q_0^2 + q_1^2 + q_3^2 + q_4^2}$

$\vec{q}_{Normalized} = \frac{\vec{q}}{|q|}$

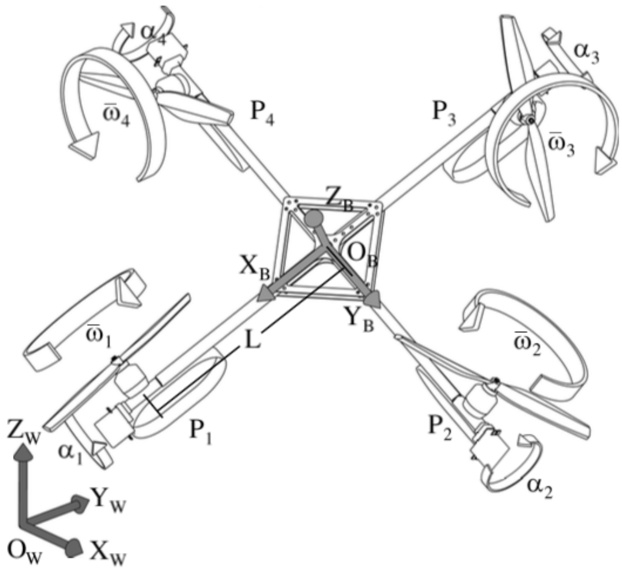


Figura 3.3: Quadrotor with tilting propellers model

3.4 Quadrotor with tilting propellers

Final model analyzed consists in an evolution of simple quadrotor, that differently from RWVTOL, try to compensate underactuation of this kind of system not adding rotors but make them able to tilt. This kind of UAV, for modellation, can be considered as a connection of 5 main rigid bodies in relative motion among themselves: the quadrotor body B and 4 propellers group P_i (multi-body system). At the top of the page is shown a picture of the model.

Formally, let's define $RF^W(O_W; X_W, Y_W, Z_W)$, as inertial reference of the world, and $RF^B(O_B; X_B, Y_B, Z_B)$ the reference frame attached to the body of the drone.
Finally is defined reference frame $RF^i(O_{Pi}; X_{Pi}, Y_{Pi}, Z_{Pi})$, attached to the propellers P_i, with X_{Pi} representing the tilting actuation axis and Z_{Pi} the propeller actuated spinning axis. In this case representation of posture is realized using rotation matrices, having R_B^W rotation matrix from RF^B to RF^W and R_i^B from RF^{Pi} to RF^B
Denoting $\alpha_i \in R$ the propeller tilt angle around axis X_{Pi}, looking the picture of the model is possible to derive the following relations:

$\mathrm{R}_{Pi}^B = R_Z((i-1)\frac{\pi}{2})\ R_X(\alpha_i)$, (i=1, 2, 3, 4)

$\vec{O}_{Pi}^B = R_Z((i-1)\frac{\pi}{2})\ (L, 0, 0)^T$, (i=1, 2, 3, 4), origin of the RF^{Pi} represented in RF^B.

So the quadrotor configuration is represented by the body position $\vec{O}_B^W$, orientation R_B^W and 4 tilt angles $\vec{\alpha}$, specifying the propeller group orientations in the body frame.

3.4.1 Dynamics equation of Quadrotor with tilting propellers

Let $\omega_B \in R^3$ be the angular velocity of the quadrotor body expressed in body frame. Given the $i-th$ propeller group P_i, the angular velocity of the propeller (in its frame), is

$\omega_{Pi} = (R_{Pi}^B)^T \omega_B + (\dot{\alpha}_i, 0, \bar{\omega}_i)^T$, where $\bar{\omega}_i \in R$ is the spinning velocity of the propeller about Z_{Pi}.

From that relation, deriving respect to time is obtained

$\dot{\omega}_{Pi} = (R_{Pi}^B)^T \dot{\omega}_B + (\dot{R}_{Pi}^B)^T \omega_B + (\ddot{\alpha}_i, 0,\ \dot{\bar{\omega}}_i)^T$

Then, applying **Euler's equation** of motion, it follow that

$\tau_{Pi} = I_{Pi}\dot{\omega}_{Pi} + \omega_{Pi} \wedge I_{Pi}\omega_{Pi} - \tau_{ext}$, with $I_{Pi} \in R^{3x3}$ is the constant inertia matrix of the propeller group and τ_{ext} is any external torque applied to the propeller.

Considering now the quadrotor body B and the four propellers P_i

$\tau_B = I_B\dot{\omega}_B + \omega_B \wedge I_B\omega_B + \sum_{i=1}^4 R_{Pi}^B \tau_{Pi}$

Therefore, for definition of force's momentum

$\tau_B = \sum_{i=1}^4 (O_{Pi}^B \wedge R_{Pi}^B T_{Pi})$

Finally, for quadrotor body position stands the following equation (**Netwon's equation**)

$\mathrm{m}\ddot{p} = \mathrm{m}(0,0,-g)^T + R_B^W \sum_{i=1}^4 R_{Pi}^B T_{Pi}$

3.4.2 Simplified model used for open loop simulation

In order to have a model that can be use for control design, some simplifications were implemented:

- Motors actuate tilting axes through microcontroller able to impose desired speeds $\omega_{\alpha_i} = \dot{\alpha}_i$ with negligible transients, not considering dynamics of motors.
- In this simplified model gyroscopic and inertial effects are at first order neglected.

So in this model, inputs are considered ω_{α_i} and $\bar{\omega}_i$.
Under the stated assumptions, final model of open loop system is

$$\begin{cases} \ddot{p} = \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} + \frac{1}{m} R_B^W F(\alpha)\omega \\ \\ \dot{\omega}_B = I_B^{-1}\tau(\alpha)\ \omega \\ \\ \dot{\alpha} = \omega_\alpha \\ \\ \dot{R}_B^W = S(\omega_B)R_B^W \end{cases}$$

with

$$F(\alpha) = \begin{pmatrix} 0 & -k_f s_2 & 0 & k_f s_4 \\ -k_f s_1 & 0 & k_f s_3 & 0 \\ k_f c_1 & -k_f c_2 & k_f c_3 & -k_f c_4 \end{pmatrix}$$

$$\tau(\alpha) = \begin{pmatrix} 0 & -Lk_f c_2 - k_m s_2 & 0 & Lk_f c_4 + k_m s_4 \\ -Lk_f c_1 + k_m s_1 & 0 & Lf_f c_3 - k_m s_3 & 0 \\ -Lk_f s_1 & Lk_f s_2 - k_m c_2 & -Lk_f s_3 - k_m c_3 & Lk_f s_4 - k_m c_4 \end{pmatrix}$$

3.4.3 Open Loop Numerical Simulation Methodology

The numerical simulation is essentially divided in the following steps:

- Firstly integration of the ω_α in order to obtain α
- Then calculation of the torque $\tau(\alpha)$ and sequentially $\dot{\omega}_B$, which integration gives back ω_B
- Using $\dot{\omega}_B$ is calculated $\dot{R}_B^W = S(\omega_B)R_B^W$

- As seen in previous simulation, after numerical simulation there is the need of the extraction of rotation matrix.
- After the extraction of the rotation matrix the linear position dynamics is approached, with the calculation of $F(\alpha)$
- Finally is obtained $\ddot{p} = R_B^W * F * \omega_B + (0, 0, -g)^T$, whose integration gives $\dot{p}$ and eventually p

Capitolo 4

Closed Loop Simulations and Results

In order to evaluate open loop systems were introduced different kind of controllers, in particular :

- For the Hexacopter simple static feedback linearization was used for height and posture regulation.
- For the Helicopter was used a linear model for the plant and the controller is just a **proportional feedback** controller for regulation of position coordinates and Euler's angles, given an arbitrary initial condition.
- Instead for the quadrotor with tilting propellers was implemented a controller using the **dynamic output linearization**, that allowed the more sophisticated task of trajectory tracking and posture tracking.

4.1 Hexacopter's height and posture control

For the Hexacopter, through inversion of z dynamics, was possible to find input that stabilizes height:

$\ddot{z} = -g + \frac{T}{m} * (q_0^2 - q_1^2 - q_2^2 + q_3^2)$

which inversion gives

$T = \frac{m(\ddot{z}+g)}{(q_0^2-q_1^2-q_2^2+q_3^2)}$

Then considering as new input $\ddot{z} = u$, and putting $u = \ddot{z}_d + K_p(z_d - z) + K_d(\dot{z}_d - \dot{z})$, that assures theoretically exponential convergence to 0 of error e_z, is obtained

T $= \frac{m(\ddot{z}_d+K_p(z_d-z)+K_d(\dot{z}_d-\dot{z})+g)}{(q_0^2-q_1^2-q_2^2+q_3^2)}$

In the same way for the torques was used an approach quite similar, defining

$\tau_\phi = K_{p_\phi}(\phi_d - \phi) + K_{d_\phi}(\dot{\phi}_d - \dot{\phi})$

$\tau_\theta = K_{p_\theta}(\theta_d - \theta) + K_{d_\theta}(\dot{\theta}_d - \dot{\theta})$

$\tau_\psi = K_{p_\psi}(\psi_d - \psi) + K_{d_\psi}(\dot{\psi}_d - \dot{\psi})$

For the simulation were used the following gains:

- $Kp_z = 0.1$, $Kd_z = 1.0$
- $K_{p_\phi} = 0.01$, $K_{p_\theta} = 0.01$, $K_{p_\psi} = 0.01$
- $K_{d_\phi} = 0.01$, $K_{d_\theta} = 0.01$, $K_{d_\psi} = 0.01$

the following initial conditions

- $x_0 = 0.0$, $y_0 = 0.0$, $z_0 = 1.0$
- $\dot{x}_0 = 0.0$, $\dot{y}_0 = 0.0$, $\dot{z}_0 = 0.0$

- $\phi_0 = 0.5$, $\theta_0 = 0.5$, $\psi_0 = 0.5$
- $\dot{\phi}_0 = 0.0$, $\dot{\theta}_0 = 0.0$, $\dot{\psi}_0 = 0.0$

and finally the following references (for z there is a trajectory tracking)

- $\phi_d = 0.0$;, $\theta_d = 0.0$;, $\psi_d = 0.0$;
- $\dot{\phi}_d = 0.0$;, $\dot{\theta}_d = 0.0$;, $\dot{\psi}_d = 0.0$;
- $z_d = cos(t)$, $\dot{z}_d = -sin(t)$, $\ddot{z}_d = -cos(t)$

In the next page are shown results for every variable controlled, so height and Euler's angles.

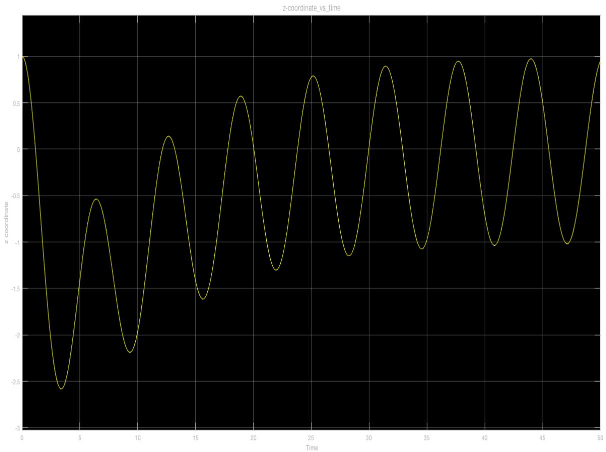


Figura 4.1: Z coordinate vs Time

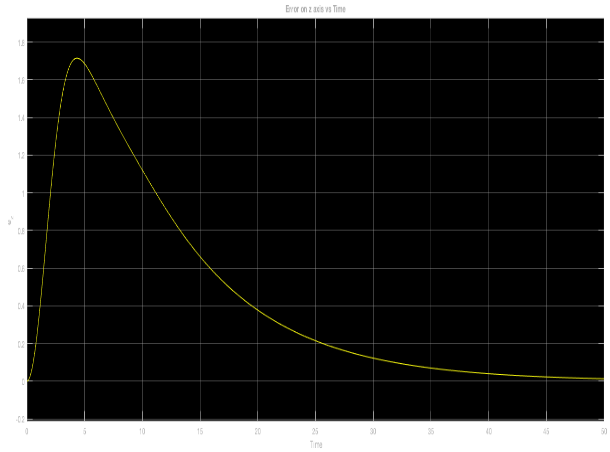


Figura 4.2: Error on Z coordinate vs Time

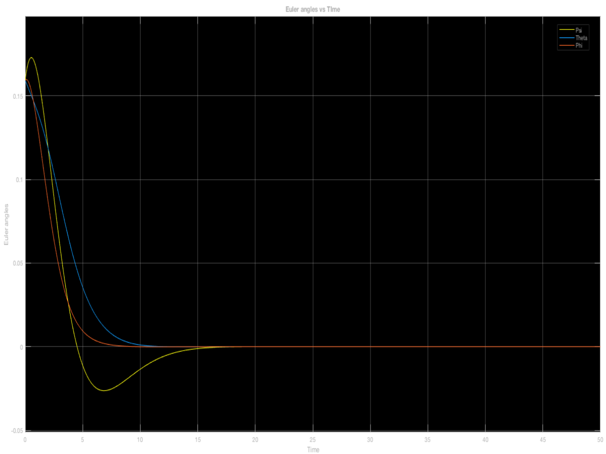


Figura 4.3: Euler's angles vs Time

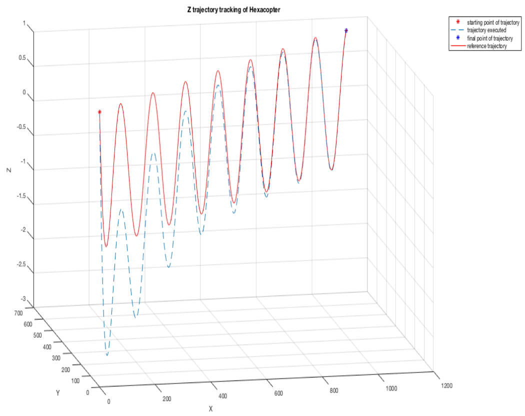


Figura 4.4: Trajectory of the Hexacopter, blue line represents the trajectory executed while the red one the reference for z axis

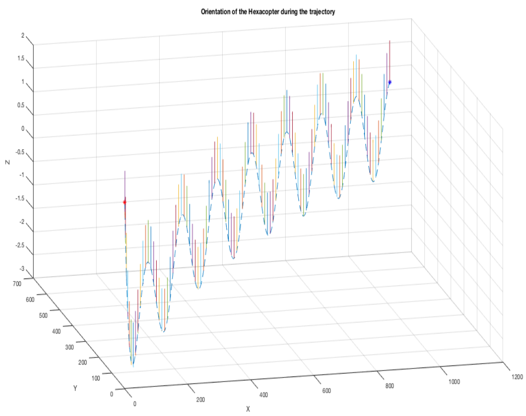


Figura 4.5: Orientation of the Hexacopter during the trajectory, axis of the drone are represented

4.2 Helicopter's position and posture control

In order to control this kind of vehicle, in close loop simulations was used a model different from the open loop one, in which dynamics is linearized. This kind of model is often used as linearized model around the *Trim* trajectory (the one with $\dot{v} = 0$ and $\dot{\omega} = 0$) and with the usage of *Euler's Angles*. By assuming that translational and rotational velocities are relatively small (near hovering), all aerodynamic drag and lift forces, which are proportional to the square of the vehicle velocities, can be neglected. In turn, the momentum drag $-Qv$ can be approximated by $Q = \frac{k_5^e}{mg}$.

Therefore, one obtains

$$\begin{pmatrix} F_B \\ \\ \Gamma_B \end{pmatrix} \approx \begin{pmatrix} -(T - mg)e_3 - \frac{1}{L}S(e_3)\Gamma - Qv \\ \\ \Gamma - \epsilon_m QS(e_3)v \end{pmatrix}$$

Denoting velocity $v_I = \dot{x}$ linear velocity in the inertial reference frame, so extending the state and transforming the second order systems in double amount of first order systems, are obtained 4 SISO system, each one couples one linear coordinate and one of the Euler angle

- **Longitudinal Channel**

$$\begin{pmatrix} \dot{x} \\ \dot{v}_{I,x} \\ \dot{\theta} \\ \dot{\omega}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{Q}{m} & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-\epsilon_m Q}{J_1} & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ v_{I,x} \\ \theta \\ \omega_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{mL} \\ 0 \\ \frac{1}{J_1} \end{pmatrix} \Gamma_2$$

- **Lateral Channel**

$$\begin{pmatrix} \dot{y} \\ \dot{v}_{I,y} \\ \dot{\phi} \\ \dot{\omega}_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{Q}{m} & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\epsilon_m Q}{J_2} & 0 & 0 \end{pmatrix} \begin{pmatrix} y \\ v_{I,y} \\ \phi \\ \omega_1 \end{pmatrix} + \begin{pmatrix} 0 \\ -\frac{1}{mL} \\ 0 \\ \frac{1}{J_2} \end{pmatrix} \Gamma_1$$

- **Altitude Channel**

$$\begin{pmatrix} \dot{z} \\ \dot{v}_{I,z} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{Q}{m} & 0 \end{pmatrix} \begin{pmatrix} z \\ v_{I,z} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} (T - mg)$$

- **Yaw Channel**

$$\begin{pmatrix} \dot{\psi} \\ \dot{\omega}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \psi \\ \omega_3 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{J_3} \end{pmatrix} \Gamma_3$$

In order to stabilize these systems, a feedback controller proportional to the state was used, with a gain matrix for each system, defining $u_i = -K_i * x_i$, with $i = 1, 2, 3, 4$.
Below are shown the gain matrices used:

- $K_1 = \begin{pmatrix} -0.0024 & -0.0191 & 0.0543 & 0.0141 \end{pmatrix}$
- $K_2 = \begin{pmatrix} 0.0024 & 0.0191 & 0.0543 & 0.0141 \end{pmatrix}$
- $K_3 = \begin{pmatrix} -0.0236 & -1.3500 \end{pmatrix}$

- $K_4 = \begin{pmatrix} 0.0200 & 0.3000 \end{pmatrix}$

and the following initial conditions :

- $\bar{x}_0 = [0.1, 0.2, 0.3]^T$
- $\bar{v}_0 = [0.0, 0.0, 0.0]^T$
- $[\phi_0, \theta_0, \psi_0]^T = [0.0, 0.2, 0.2]^T$
- $\bar{\omega}_0 = [0.0, 0.0, 0.0]^T$

Using those parameters, are obtained the following results (next page) which show asymptotic stabilization to $\bar{0}$ of the state :

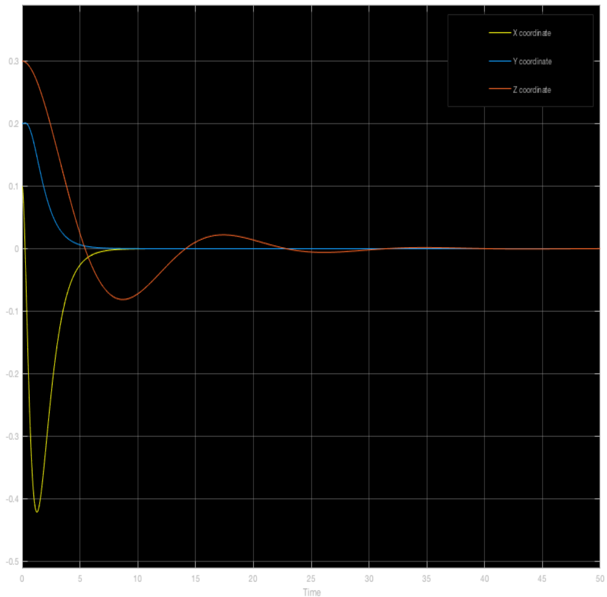


Figura 4.6: Linear Coordinates of the linearized helicopter vs Time

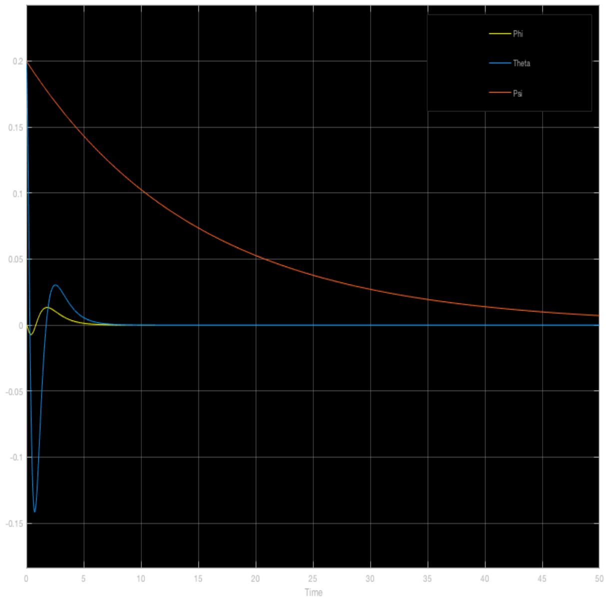


Figura 4.7: Euler Angles of the linearized helicopter vs Time

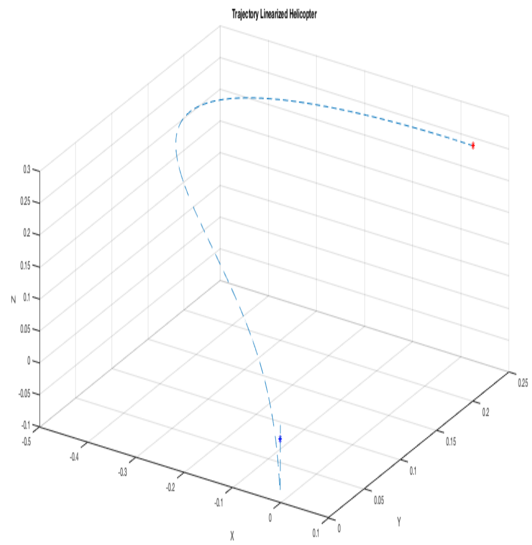


Figura 4.8: Trajectory of the linearized helicopter, red dot is the start position, blue dot is the last position

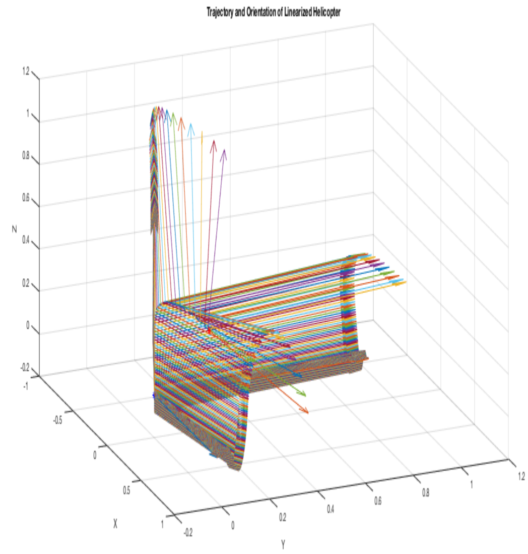


Figura 4.9: Orientation of the linearized helicopter, axis of the helicopter are shown during the motion

4.3 Quadrotor control for trajectory tracking

The model of the quadrotor with tilting propellers is non-linear and moreover is not possible to realize a simple input-out linearization (as the hexacopter) in order to control height and orientation.The inversion has to be realized at higher differential level compared to input ω.
Formally, let's start from the dynamics equations for the simplified model previously expressed:

$$\begin{cases} \ddot{p} = \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} + \frac{1}{m} R_B^W F(\alpha)\omega \\ \\ \dot{\omega}_B = I_B^{-1} \tau(\alpha)\ \omega \\ \\ \dot{\alpha} = \omega_\alpha \\ \\ \dot{R}_B^W = S(\omega_B) R_B^W \end{cases}$$

Is possible to rewrite the that system as :

$$\begin{pmatrix} \ddot{p} \\ \dot{\omega}_B \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \\ \dot{\omega}_B \end{pmatrix} + \begin{pmatrix} \frac{1}{m} R_B^w & 0 \\ 0 & I_B^{-1} \end{pmatrix} \begin{pmatrix} F(\alpha) & 0 \\ \tau_\alpha & 0 \end{pmatrix} \begin{pmatrix} \omega \\ \omega_\alpha \end{pmatrix}$$

$$= f + J_R \begin{pmatrix} \bar{J}_\alpha & 0 \end{pmatrix} \begin{pmatrix} \omega \\ \omega_\alpha \end{pmatrix} = f + J_R J_\alpha(\alpha) \begin{pmatrix} \omega \\ \omega_\alpha \end{pmatrix}$$

$$= f + J(\alpha) \begin{pmatrix} \omega \\ \omega_\alpha \end{pmatrix}$$

The matrix $J(\alpha)$, *Output Jacobian*, is a 6x8 matrix, that could be inverted using generalized inverse(like pseudo-inverse). The main problem in this approach is the fact that $\rho(J) = rank(J) = rank(J_R J_\alpha) = rank(J_\alpha)$, since J_R is a not-singular square matrix, but $rank(J_\alpha) \leq 4 < 6$ because of the structural null matrix $\mathbf{O} \in R^{6x4}$.
A systematic way to approach this kind of problem is the *Dynamic Output Linearization*, trying to to invert the input-output map at a higher differential level where inputs ω_α will explicitly appear.

Formaly, let's expand the term

$J_\alpha(\alpha)\omega = \sum_{i=1}^{4} j_{i,\alpha}\omega_i$

and noting that

$\frac{d}{dt} J_\alpha(\alpha)\omega = J_R J_\alpha(\alpha)\dot{\omega} + J_R \sum_{i=1}^{4} \frac{d}{d\alpha} j_{i,\alpha}\omega_\alpha\omega_i + \dot{J}_R J_\alpha(\alpha)\omega$

Differentiation of the dynamics equation brings then to

$$\begin{pmatrix} \dddot{p} \\ \\ \ddot{\omega}_B \end{pmatrix} = J_R \begin{pmatrix} J_\alpha(\alpha) & \sum_{i=1}^{4} \frac{d}{d\alpha} j_{i,\alpha}\omega_\alpha\omega_i \end{pmatrix} \begin{pmatrix} \dot{\omega} \\ \\ \omega_\alpha \end{pmatrix} + \begin{pmatrix} \frac{R_B^W}{m} F(\alpha)\omega \\ \\ 0 \end{pmatrix}$$

$$= A(\alpha, \omega) \begin{pmatrix} \dot{\omega} \\ \\ \omega_\alpha \end{pmatrix} + b(\alpha, \omega, \omega_B)$$

Then assuming that $rank(A) = rank_{max} = 6$, is possible to invert the relation using generalized inverse (in the simulation was used pseudo-inverse) having eventually

$$\begin{pmatrix} \dot{\omega} \\ \\ \omega_\alpha \end{pmatrix} = A^{+}(\begin{pmatrix} \dddot{p}_r \\ \\ \ddot{\omega}_r \end{pmatrix} - b) + (I_8 - A^{+}A)z$$

with second part of the expression that projects vector z onto the 2D null space of A. This allows to achieve full input-output linearization

$$\begin{pmatrix} \dddot{p} \\ \\ \ddot{\omega}_B \end{pmatrix} = \begin{pmatrix} \dddot{p}_r \\ \\ \ddot{\omega}_r \end{pmatrix}$$

Now in order to solve the original control problem, assuming that $p_d \in C^3$, is sufficient to set

$$\dddot{p}_r = \dddot{p}_d + K_{p_1}(\ddot{p}_d - \ddot{p}) + K_{p_2}(\dot{p}_d - \dot{p}) + K_{p_3}(p_d - p)$$

This allows exponential and decoupled convergence of the positional error to $\bar{0}$. However there is then the need of control the orientation; the solution depends from the representation used for the SO(3) group. In this case were used the rotation matrices parametrization, defining also an orientation error. Assuming also that $R_d \in C^3$, and reminding the $[\cdot]_\vee$ that represents the inverse map from SO(3) to R^3, is possible to define the orientation error

$$e_r = \frac{1}{2}[(R_B^W)^T R_d - R_d^T R_B^W]_\vee$$

and the desired $\omega_d = [R_d^T \dot{R}_d]$, that in our case is just $\bar{0}$, since for posture was set a regulation problem (R_d is a constant and $\dot{R}_d = 0_3$).

After that the orientation error and the ω_d has be defined, is possible to choice

$$\ddot{\omega}_r = \ddot{\omega}_d + K_{\omega_1}(\dot{\omega}_d - \dot{\omega}) + K_{\omega_2}(\omega_d - \omega) + K_{\omega_3}(e_r)$$

Despite of, there is still a redundancy (the projection in the null space of A), that can be exploited to achieve other tasks. However in this case this property wasn't used.
Below are shown the gain matrices used:

- $K_{p_1} = I_3 * 7.5$
- $K_{p_2} = I_3 * 25.75$
- $K_{p_3} = I_3 * 15$
- $K_{\omega_1} = I_3 * 3$
- $K_{\omega_2} = I_3 * 3$
- $K_{\omega_3} = I_3 * 10$

the initial conditions:

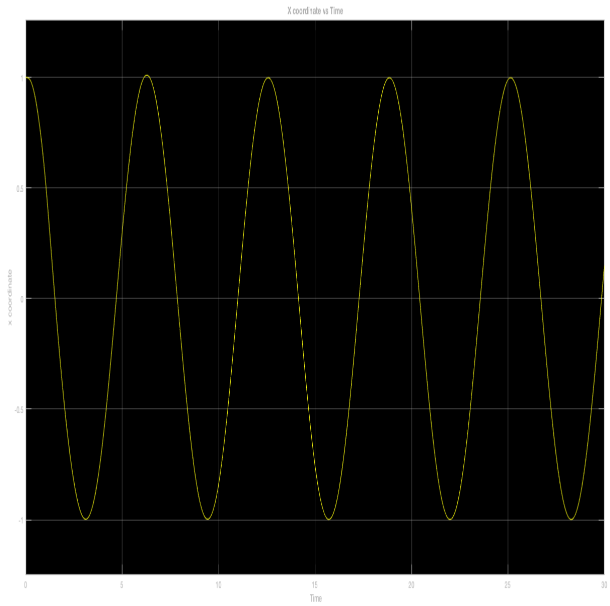
- $\bar{x}_0 = [1.0, 0.0, 0.0]^T$
- $\bar{v}_0 = [0.0, 0.0, 1.0]^T$
- $\mathrm{R}_{B0} = I_3$
- $\bar{\omega}_0 = [0.0, 0.0, 0.0]^T$

and the reference for the controller

- $x_d = cos(t), \dot{x}_d = -sin(t), \ddot{x}_d = -cos(t), \dddot{x}_d = sin(t)$
- $y_d = 0.0, \dot{y}_d = 0.0, \ddot{y}_d = 0.0, \dddot{y}_d = 0.0$
- $x_d = sin(t), \dot{x}_d = cos(t), \ddot{x}_d = -sin(t), \dddot{x}_d = -cos(t)$

- $R_d = \begin{pmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.9950 & 0.0998 \\ 0.0000 & -0.0998 & 0.9950 \end{pmatrix}$, corresponding a rotation around x axis of 0.1 radians.

In the next page are shown results of the simulations with given parameters.



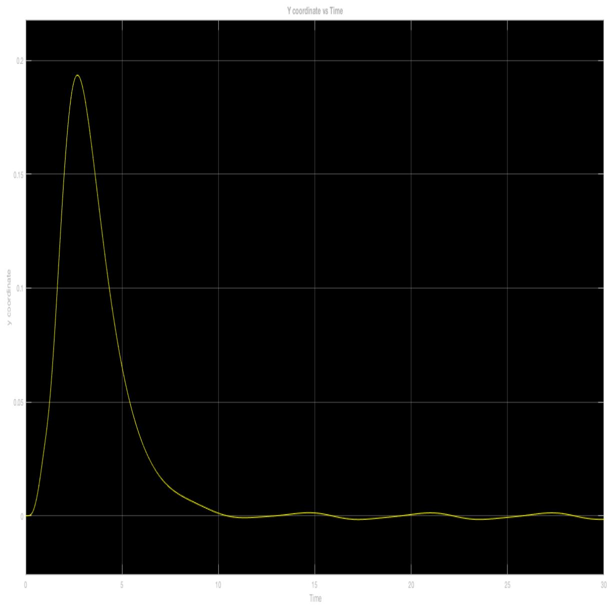
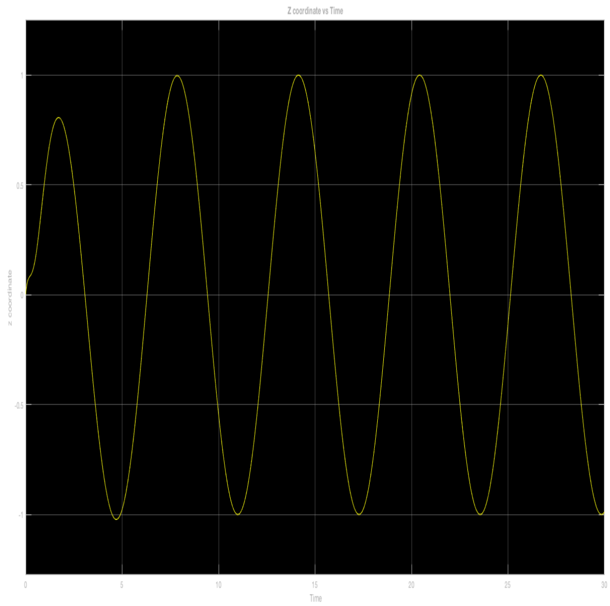


Figura 4.10: X coordinate of the quadrotor vs Time

Figura 4.11: Y coordinate of the quadrotor vs Time



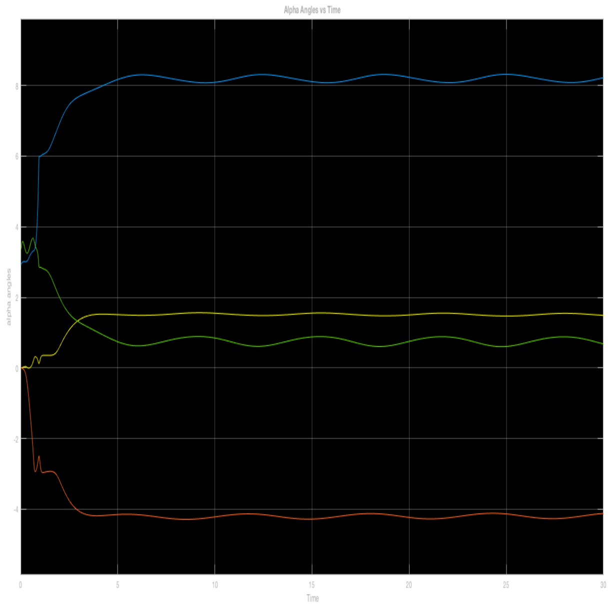
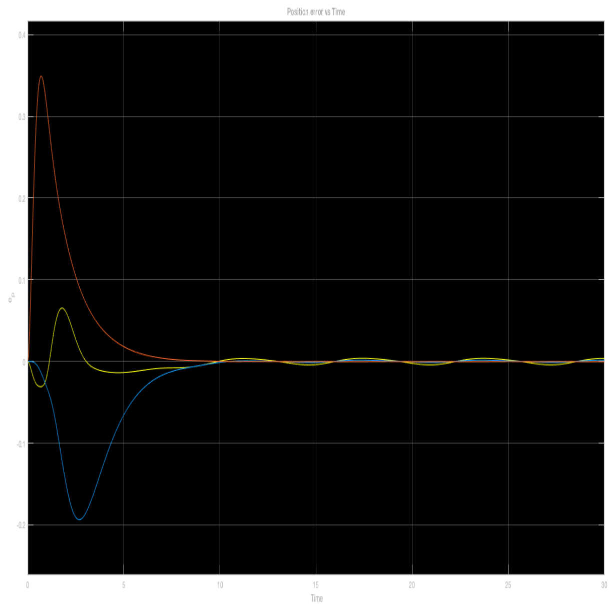


Figura 4.12: Z coordinate of the quadrotor vs Time

Figura 4.13: Alpha angles vs Time



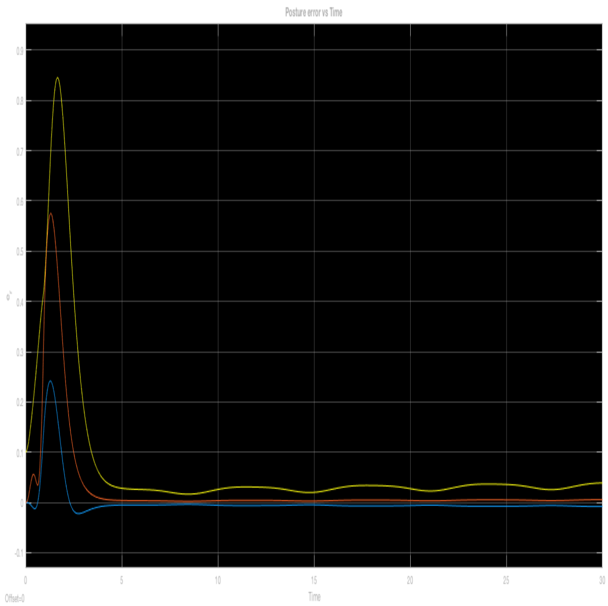


Figura 4.14: Position error vs Time

Figura 4.15: Posture error vs Time

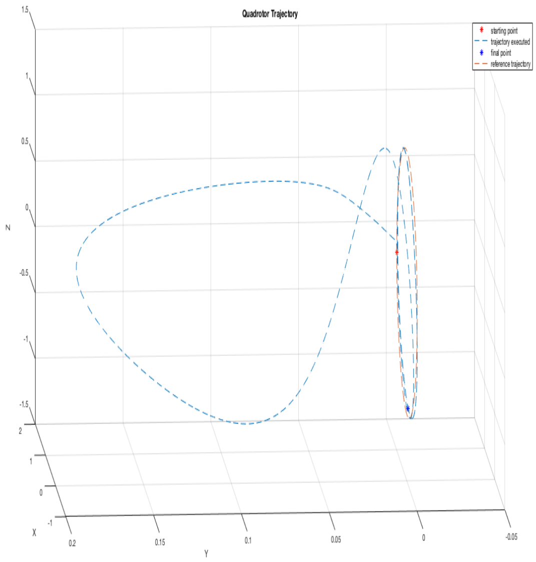


Figura 4.16: Trajectory of the Quadrotor, blue line representes the executed trajectory, the red one the reference trajectory

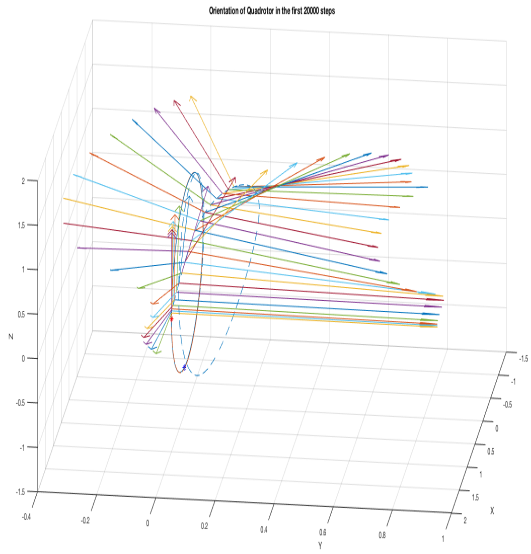


Figura 4.17: Posture of the Quadrotor during the trajectory, represented for the first 20000 steps

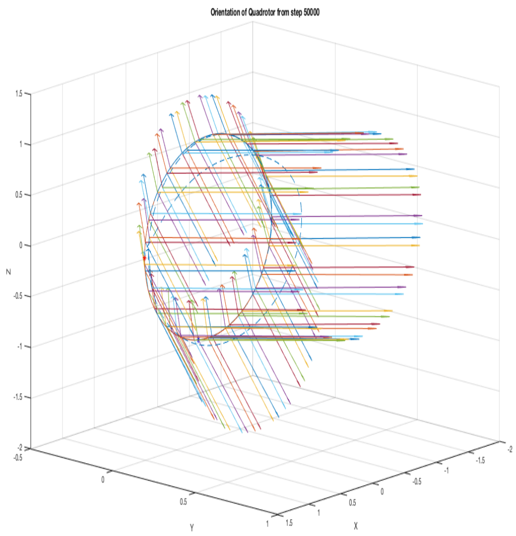


Figura 4.18: Posture of the Quadrotor during the trajectory, represented from the 50000 step until the end

Capitolo 5

Conclusion

The project that has been developed had as main goal the study and simulations of not so usual kind of RWVTOLs, compared with the classic quadrotor's model. The checking of the models validity needed the implementation of different controllers, in order to verify the models consistency. Both closed loop and open loop simulations have exalted the common issues that come out in that context, as

- posture representation, integration and the use of different kind of parameterizations
- dynamics inversion for input output linearization, either for static and dynamics feedback linearization
- Gains tuning of controllers

Despite all, according to results is possible to appreciate very good convergence of all these simulations, also with basic controllers (as for Helicopter and Hexacopter). However, all this simulations were realized not considering aerodynamic effects (only the helicopter consider an approximation of the momentum drag in closed loop simulation), external disturbances and too simple modeling to face up with tests with real hardware.
Possible further developments that can start from this project may be

- models improvement, including also not considered effects
- implementation of more complicated controllers (as geometric controllers or optimal controllers)
- eventually, testing on real hardware

Capitolo 6

Bibliography

- "Robotics, Modelling, Planning and Control", B. Siciliano, L.Sciavicco, L.Villani, G.Oriolo
- "Contributions to the automatic control of aerial vehicles", Minh Duc Hua
- "Modeling and Control of a Quadrotor UAV with Tilting Propellers", M.Ryll, H.Heinrich Bulthoff, P. R. Giordano
- "Mathematical Modeling and Control of a Hexacopter", A. Alaimo, V. Artale, C. Milazzo, A. Ricciardello, L. Trefiletti
- "Modelling Dynamic Systems - Matlab Simulink", Mathworks Italia, "https://it.mathworks.com/help/simulink/ug/modeling-dynamic-systems.html"
- "Progettazione e analisi di sistemi di controllo - Control System Toolbox", Mathworks Italia, "https://it.mathworks.com/products/control.html"