DESIGN AND NAVIGATION OF A TERRESTRIAL AND AERIAL HYBRID VEHICULE

Josue Colmenares Vazquez

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : AUTOMATIQUE - PRODUCTIQUE

Arrêté ministériel : 25 mai 2016

Présentée par

JOSUE COLMENARES VAZQUEZ

Thèse dirigée par **Nicolas MARCHAND**, CNRS
et codirigée par **Pedro CASTILLO**

préparée au sein du **Laboratoire Grenoble Images Parole Signal Automatique**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)**

Conception et navigation d'un véhicule hybride terrestre et aérien

DESIGN AND NAVIGATION OF A TERRESTRIAL AND AERIAL HYBRID VEHICULE

Thèse soutenue publiquement le **28 juin 2018**,
devant le jury composé de :

Monsieur NICOLAS MARCHAND
DIRECTEUR DE RECHERCHE, CNRS DELEGATION ALPES, Directeur de thèse
Monsieur MOHAMED BOUTAYEB
PROFESSEUR, UNIVERSITE DE LORRAINE, Président
Monsieur FRANCK RUFFIER
CHARGE DE RECHERCHE, CNRS DELEGATION PROVENCE ET CORSE, Examinateur
Monsieur PEDRO GARCIA-GIL
MAITRE DE CONFERENCES, UNIVERSITE DE VALENCE - ESPAGNE, Rapporteur

UNIVERSITÉ DE GRENOBLE ALPES

ÉCOLE DOCTORALE EEATS

Électronique Électrotechnique Automatique et Traitement du signal

T H È S E

pour obtenir le titre de

docteur en sciences

de l'Université de Grenoble Alpes

Mention : Automatique Productique

Présentée et soutenue par

Josué COLMENARES-VAZQUEZ

Conception et navigation d'un véhicule hybride terrestre et aérien

Thèse dirigée par Nicolas MARCHAND
et co-dirigée par Pedro CASTILLO

préparée au laboratoire Grenoble Images Parole Signal Automatique (GIPSA-lab)

soutenue le 28 juin 2018

Jury :

Directeur :	Monsieur NICOLAS MARCHAND	-	GIPSA-Lab
Président :	Monsieur MOHAMED BOUTAYEB	-	Université de Lorraine
Examinateur :	Monsieur FRANCK RUFFIER	-	Aix Marseille Université
Rapporteur :	Monsieur PEDRO GARCIA-GIL	-	Université Politechnique de Valence

Acknowledgments

This thesis began with the full support of my loved wife Dulce and was possible thanks to her patience and encouragement she gave me during the development of this work. What would I do without you? I owe you so much and I really thank you with all my heart.

I am very grateful with my mom Erindelia for all her pieces of advice and her blessings. I want to share this achievement with you and with all my brothers: Andrés, Edgar and Roni.

Also, I want to express my sincerely gratitude to my thesis directors. First of all, for accepting me in this project and for all their pieces of advice and their guidelines all along this work. I very appreciated all their comments, corrections and assertions, I know they tried to give their best in order to develop this thesis in the best way possible. Thanks a lot Nicolas, Pedro and José Ernesto.

I express my absolutely acknowledge to each member of my defense jury. Thanks for accepting to review my work, for taking your time to analyze my manuscript and for your highly valuable remarks and comments. I really want you to know I can not thank you enough for all your efforts to come to my defense in spite of the difficulties in the transport. Thanks so much Mohamed Boutayeb, Frank Ruffier and Pedro Garcia-Gil.

I have to say all my stay at the Gipsa laboratory were in a very agreeable atmosphere and in the company of good friends and excellent people, over qualified administrative staff, very competent technicians, brilliant engineers and exceptional researchers. I would like to thank my mates of the UAV team for all the moments shared all the while this thesis was developed, thanks Jonatan, José Juan and Bruno. Also, thanks to all the PhD fellows for all the coffee breaks at the afternoons that let me know them and appreciate their life experiences.

Marie-Jo, I truly appreciate all the hours you passed with me teaching me French and the french culture at the association Coup de Pouce in Grenoble. Thank to all the staff of Coup de Pouce for their outstanding dedication and attention given to the student community. I hope you will be rewarded.

Finally, I especially thank to Conacyt of Mexican government for financing and making possible this work. Thanks for believing in young people and for helping their dreams to become true.

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

UGV	*Unmanned Ground Vehicle*
UAV	*Unmanned Aerial Vehicle*
iff	*if and only if*
DALER	*Deployable Air Land Exploration Robot*
VTOL	*Vertical Take-Off and Landing*
HYTAQ	*Hybrid Terrestrial and Aerial Quadrotor*

Chapter 1

State of the Art

Contents

1.1 Overview

Quadrotor was a technological breakthrough because it gives the possibility of flying in order to make tasks that were not possible with terrestrial vehicles. But the main drawback until now is the operation time which not exceed more than 30 minutes at best. One possible solution might come from the advances on the energy stocking technology in batteries. Another one is to conceive new kind of vehicle more efficient and more adaptable to complex situations. From this, it rises the need of hybrid vehicles capable to overcome energy issues and also having a greater adaptability. During the conception of these new vehicles is also necessary to design the control algorithm for position or tracking in every operation mode. The conception of a vehicle must be in such a way the performance of each locomotion mode be equal or almost equal to a vehicle with a single locomotion strategy. In this section, an overview of some outstanding present prototypes is covered and followed by a brief revision of elementary concepts involved in the modeling of a rigid body.

1.2 Hybrid Vehicles Prototypes around the World

There is no a large amount of vehicles with a hybrid structure in the literature. Most of them, only exploit one locomotive ability, that is, rolling, walking, flying, hovering, climbing, swimming, crawling, or jumping. With technology evolution, every day the needs become more complex and with more specific requirements. Vehicles need to be more flexible and adaptable to different situations and environments. The tasks could cover a wide spectrum like rescue activities, surveillance, mapping, exploration in risky areas, inspection in complex environments as forests, semi-collapsed buildings, deep caverns, among others. These present requirements begins to motivate the study, conception and developing of new kind of vehicles that could accomplish a larger dimension of specifications. This work focuses in the conception and developing of an aerial and terrestrial vehicle capable of flying and moving over ground. One of the prototype that exists in present day is the DALER (Deployable Air Land Exploration Robot) [1]. This vehicle is inspired from the nature. For example, a salamander can adopt a wheel shape and roll down in order to scape from its predators. In such a way, this vehicle behaving firstly as a VTOL (vertical take-off and landing) can use its wings to walk over ground in short distances. Reusing the same structure allows to minimize the structural mass of the robot and to reduce the drawback of performance that comes with additional parts. The obstacles found by terrestrial vehicles are easily overcome flying and not represent anymore an issue.

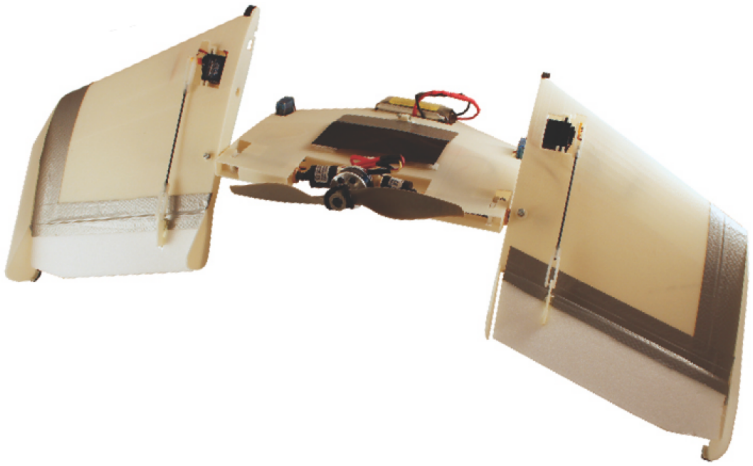


Figure 1.1: Prototype of the Deployable Air Land Exploration Robot (DALER) a flying and walking robot that uses its wings as whegs to move on the ground.

Another platform developed by Thorel *et al.* [2] moves over ground using the four little supports. This vehicle is a quadrotor suitable for indoor exploration because of its maneuverability and its hover possibility. This prototype improves the operation time of the quadrotor because in terrestrial mode it consumes significantly less energy than in aerial mode. Therefore, the vehicle is conceived to displace on ground and to fly only when necessary, for example, to avoid obstacles such as stairs, steps or desks. The authors focused in the developing of a position control algorithm with a flatness approach.

Figure 1.2: Picture of the platform Hybrid quadrotor structure developed by Thorel *et al.*

In [3], the authors conceived a hybrid vehicle based on a quadrotor inside a lightweight spherical exoskeleton . They make an analysis of the contact point of this cage for generating the control algorithm for ground displacement. This is a small vehicle about 35 gr. with a payload of 10 gr. The portability of this vehicle could be useful in pipeline inspections, mapping and surveillance tasks. The tests show their vehicle is 260 percent more efficient when it rolls to cover the same distance than in aerial mode. The performance reduction in flight mode reduces 39 percent with the exoskeleton than without it. This vehicle is mainly designed to increase the energy efficiency.

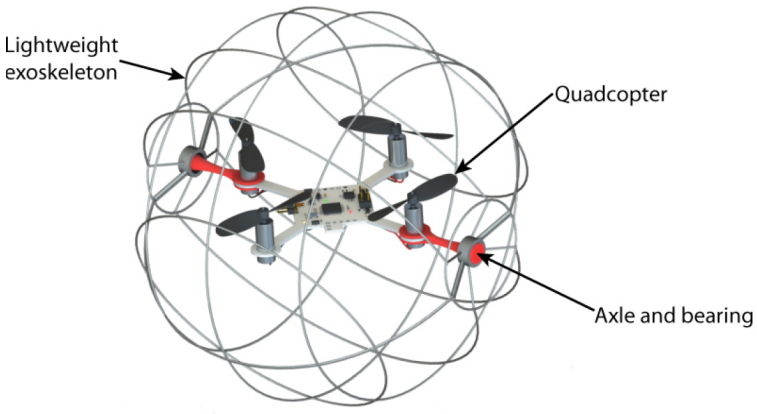


Figure 1.3: Spherical-shaped aerial terrestrial robot developed by Dudley.

Another interesting prototype was developed by Kalantari *et al.* in [4], this vehicle is a quadrotor inside a cage. The cage allows the quadrotor to roll over ground without additional actuators. The vehicle needs only to overcome the rolling resistance in terrestrial mode. Their results demonstrate the HYTAQ (Hybrid Terrestrial and Aerial Quadrotor) can operate six times longer and travel a distance four times greater than the aerial only system. In flight

mode, the vehicle behaves as a conventional quadcopter. The authors developed a control law for ground mode and focus in increasing the operation time and avoiding obstacles by flying.

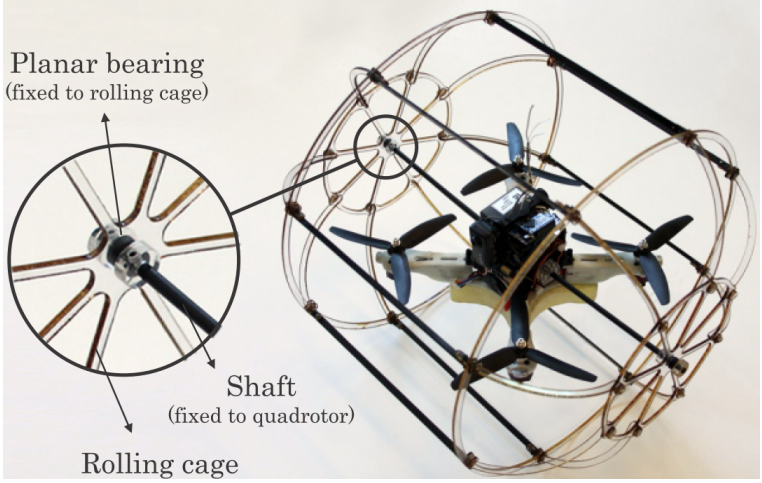


Figure 1.4: The HyTAQ, Hybrid Terrestrial and Aerial Quadrotor.

There is another work [5] called the Morphing Micro Air-Land Vehicle (MMALV) that consists of a basic fixed-wing system coupled with a wheel-leg (wheg) structure. The whegs are actuated separately from the actuator that provides the thrust for aerial movement. Another bio-inspired platform named DASH+Wings [6] uses wings flapping to enhance its locomotion when walking or climbing ramps.

The new prototypes seem to converge to a hybrid vehicle using a quadrotor with an additional passive structure to behave as a little car. On these basis, our proposed prototype does not have additional servos to the four motors of a quadrotor and possesses two passive wheels in order to move over ground. These wheels are acting also as a protection structure of the vehicle and reducing the contact points with ground with as a consequent increase in energy efficiency. These wheels represent a smaller structure compared to the cage or wings used to roll in precedent prototypes. Also, the air flow passing through helices is less perturbed than using a cage. Fast displacement are guaranteed using wheels than using feet. Fig. 1.5 shows the design of this prototype.

The remaining challenge is to design a control algorithm for its displacement over ground and in air that let us to pass from one operation mode to another smoothly.

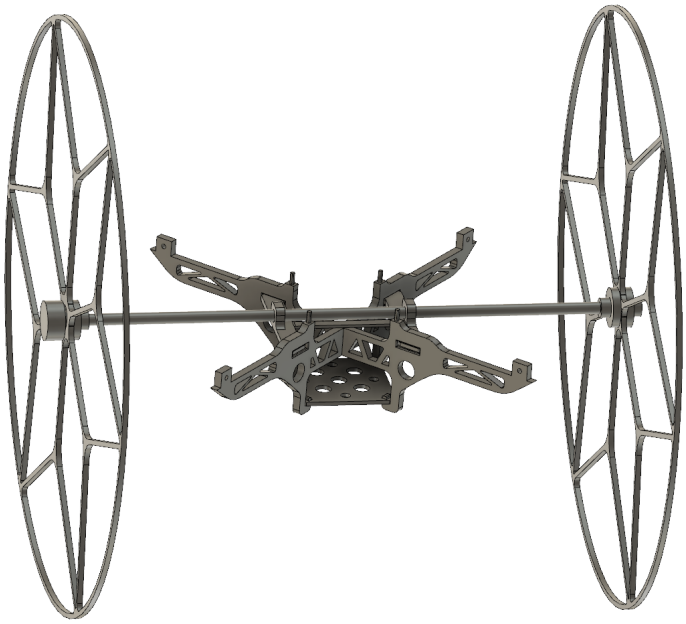


Figure 1.5: Design of the hybrid vehicle prototype.

1.3 Elementary Background

1.3.1 Inertia Moments

The inertia moment of a body represents a resistance to change its velocity of rotation. If a body has mass, so it has also a moment of inertia relative to some axis. Let us determine the inertia moment I_e around the axis $\boldsymbol{e}$ as shown in the figure 1.6.

$$I_e = \sum_{i \in C} m_i {h_i}^2$$

where

$${h_i}^2 = \boldsymbol{r}_i^T \boldsymbol{r}_i - \boldsymbol{r}_i^T \boldsymbol{e}$$

and with

$$\boldsymbol{r}_i = (x_i, y_i, z_i)^T$$

$$\boldsymbol{e} = (\alpha, \beta, \gamma)^T \qquad \|\boldsymbol{e}\| = 1$$

Therefore,

$${h_i}^2 = {x_i}^2 + {y_i}^2 + {z_i}^2 - \alpha^2 {x_i}^2 - \beta^2 {y_i}^2 - \gamma^2 {x_i}^2 - 2\alpha\beta x_i y_i - 2\beta\gamma y_i z_i - 2\alpha\gamma x_i z_i$$

and provided

$$\alpha^2 + \beta^2 + \gamma^2 = 1$$

$${h_i}^2 = \alpha^2({y_i}^2 + {z_i}^2) + \beta^2({x_i}^2 + {z_i}^2) + \gamma^2({x_i}^2 + {y_i}^2) - 2\alpha\beta x_i y_i - 2\beta\gamma y_i z_i - 2\alpha\gamma x_i z_i$$

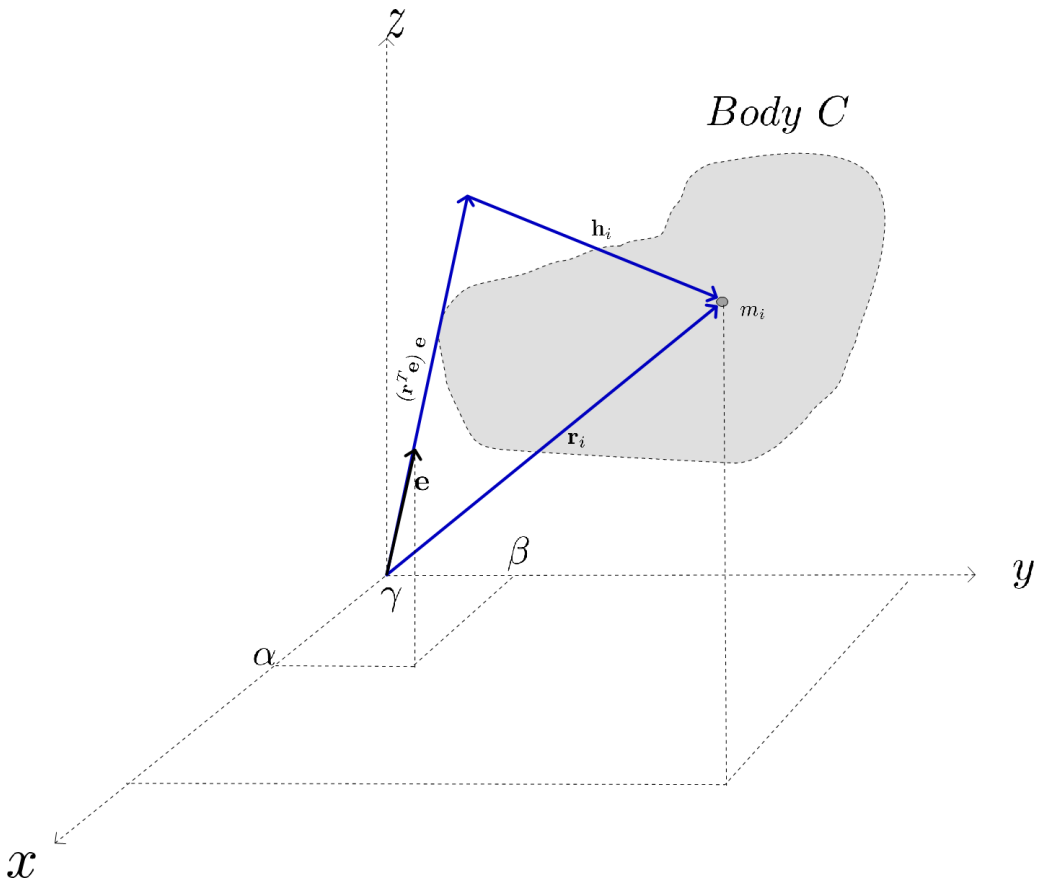


Figure 1.6: Diagram of rigid body for the computing of inertia moment about $\boldsymbol{e}$ axis.

Consequently,

$$\begin{aligned} I_e &= \alpha^2 \sum_{i \in C} m_i({y_i}^2 + {z_i}^2) + \beta^2 \sum_{i \in C} m_i({x_i}^2 + {z_i}^2) + \gamma^2 \sum_{i \in C} m_i({x_i}^2 + {y_i}^2) \\ &\quad -2\alpha\beta \sum_{i \in C} m_i x_i y_i - 2\beta\gamma \sum_{i \in C} m_i y_i z_i - 2\alpha\gamma \sum_{i \in C} m_i x_i z_i \end{aligned}$$

The following expressions are called main inertia moments:

$$I_{xx} := \sum_{i \in C} m_i({y_i}^2 + {z_i}^2) \quad I_{yy} := \sum_{i \in C} m_i({x_i}^2 + {z_i}^2) \quad I_{zz} := \sum_{i \in C} m_i({x_i}^2 + {y_i}^2)$$

and the quantities:

$$I_{xy} := \sum_{i \in C} m_i x_i y_i \quad I_{xz} := \sum_{i \in C} m_i x_i z_i \quad I_{yz} := \sum_{i \in C} m_i y_i z_i$$

They are named centrifugal inertia moments. Then, the inertia moment can be rewritten as:

$$I_e = \boldsymbol{e}^T \mathbb{I} \boldsymbol{e}$$

with

$$\mathbb{I} = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{pmatrix}$$

$$I_{xy} = I_{yx} \quad I_{xz} = I_{zx} \quad I_{yz} = I_{zy}$$

$\mathbb{I}$ is named the inertia tensor. This inertia tensor is useful to compute the kinectic energy $T_{rel,O}$ and the impulse moment $K_{rel,O}$ of a body rotating around a pivot O:

$$K_{rel,O} = \mathbb{I}\boldsymbol{w} \qquad T_{rel,O} = \frac{1}{2}\boldsymbol{w}^T\mathbb{I}\boldsymbol{w}$$

where $\boldsymbol{w}$ is the angular velocity of the body.

1.3.2 Skew Symmetric Matrix

The vectorial product between tow vectors $\boldsymbol{a}$ and $\boldsymbol{b}$,

$$\boldsymbol{c} = \boldsymbol{a} \times \boldsymbol{b}$$

can be represented as a product between a matrix and a vector:

$$\boldsymbol{c} = \mathbb{A}\boldsymbol{b}$$

where $\mathbb{A}$ will be obtained from the vector $\boldsymbol{a}$. For instance,

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \Longrightarrow \mathbb{A} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$$

Consequently, the vectorial product with the angular velocity $(\boldsymbol{\omega} \times \cdot)$ can be expressed by a matrix $\mathbb{W}$:

$$\mathbb{W} = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix}$$

This matrix is named the skew symmetric matrix.

1.3.3 Change of Coordinates

Let be two coordinated systems S_1 and S_2 and a matrix $\mathbb{T}$ such that any vector $\boldsymbol{r}$ whose coordinates are expressed in S_2 can be expressed in S_1 as:

$$\underset{S_1}{\boldsymbol{r}} = \mathbb{T}\underset{S_2}{\boldsymbol{r}}$$

The matrix $\mathbb{T}$ is a transformation matrix. The coordinated systems will be defined by their vectorial basis,

$$S_1 = (\boldsymbol{i}_1, \boldsymbol{j}_1, \boldsymbol{k}_1) \qquad S_2 = (\boldsymbol{i}_2, \boldsymbol{j}_2, \boldsymbol{k}_2)$$

Let suppose the coordinates of each basis vector of system S_2 are known and thus the transformation matrix can be written as,

$$T = \begin{pmatrix} \boldsymbol{i}_2^T\boldsymbol{i}_1 & \boldsymbol{j}_2^T\boldsymbol{i}_1 & \boldsymbol{k}_2^T\boldsymbol{i}_1 \\ \boldsymbol{i}_2^T\boldsymbol{j}_1 & \boldsymbol{j}_2^T\boldsymbol{j}_1 & \boldsymbol{k}_2^T\boldsymbol{j}_1 \\ \boldsymbol{i}_2^T\boldsymbol{k}_1 & \boldsymbol{j}_2^T\boldsymbol{k}_1 & \boldsymbol{k}_2^T\boldsymbol{k}_1 \end{pmatrix}$$

1.3.4 Rotation Matrix

A rotation matrix is a matrix $\mathbb{R}$ such that:

$$|\mathbb{R}\boldsymbol{r}| = |\boldsymbol{r}|$$

A rotation matrix satifies the following properties:

P1 $\mathbb{R}^T = \mathbb{R}^{-1}$

P2 $|\det(\mathbb{R})| = 1$

P3 The eigenvalues λ_i of $\mathbb{R}$ are:

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= \cos(\theta) + i\sin(\theta) \\ \lambda_2 &= \cos(\theta) - i\sin(\theta) \end{aligned}$$

with $\cos(\theta) = \dfrac{\mathrm{tr}(\mathbb{R}) - 1}{2}$

where $\mathrm{tr}(\mathbb{R})$ is the trace of the rotation matrix. Let us suppose the new basis vectors is known and obtained after the application of a rotation, that is:

$$S_2 = (\boldsymbol{i}_2, \boldsymbol{j}_2, \boldsymbol{k}_2)$$

where:

$$\begin{aligned} \boldsymbol{i}_2 &= \mathbb{R}\boldsymbol{i} \\ \boldsymbol{j}_2 &= \mathbb{R}\boldsymbol{j} \\ \boldsymbol{k}_2 &= \mathbb{R}\boldsymbol{k} \end{aligned}$$

therefore, the matrix rotation $\mathbb{R}$ can be expressed as:

$$\mathbb{R} = \begin{pmatrix} \boldsymbol{i}_2^T\boldsymbol{i}_1 & \boldsymbol{j}_2^T\boldsymbol{i}_1 & \boldsymbol{k}_2^T\boldsymbol{i}_1 \\ \boldsymbol{i}_2^T\boldsymbol{j}_1 & \boldsymbol{j}_2^T\boldsymbol{j}_1 & \boldsymbol{k}_2^T\boldsymbol{j}_1 \\ \boldsymbol{i}_2^T\boldsymbol{k}_1 & \boldsymbol{j}_2^T\boldsymbol{k}_1 & \boldsymbol{k}_2^T\boldsymbol{k}_1 \end{pmatrix}$$

1.3.4.1 Rotations about Axes x, y and z

Let us have a rotation about the axis $\boldsymbol{x}$ as shown in Fig. 1.7, Then, applying the expression for $\mathbb{R}$, it yields:

$$\mathbb{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix}$$

In a similar way, the rotations about $\boldsymbol{y}$ and $\boldsymbol{z}$

They yield:

$$\mathbb{R}_y = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \mathbb{R}_z = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \sin(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

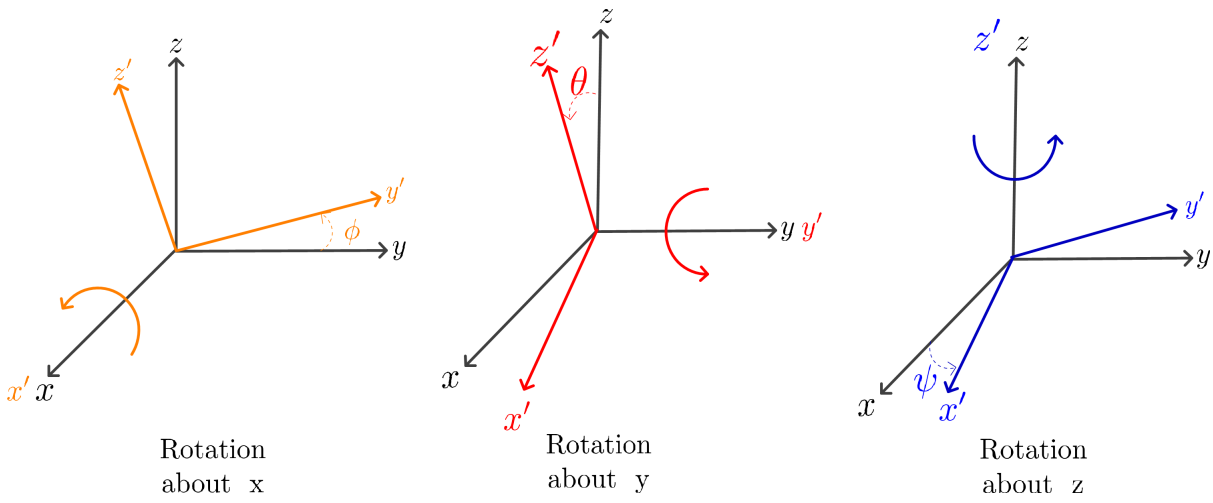


Figure 1.7: Rotations about xyz axes.

1.3.4.2 Composition of Rotations

Let us have two rotation matrices $\mathbb{R}_1$ and $\mathbb{R}_2$, so it is possible to replace this two matrix by only one rotation matrix $\mathbb{R}$.

Rotations in the Same Coordinated System In this case, the second rotation is applied in the same frame than the first one, see Fig 1.8. Then, it results,

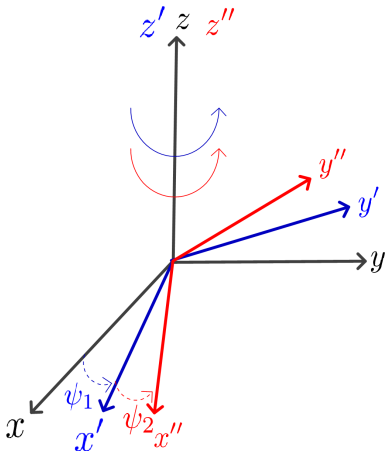


Figure 1.8: Rotations in the same framework.

$$\boldsymbol{r}_3 = \mathbb{R}_2\boldsymbol{r}_2 = \mathbb{R}_2(\mathbb{R}_1\boldsymbol{r})$$

If $\boldsymbol{r}_3 = \mathbb{R}\boldsymbol{r}$, then $\mathbb{R} = \mathbb{R}_2\mathbb{R}_1$.

Rotations in Different Coordinated Systems In this case, there are two rotation matrices $\mathbb{R}_1$ and $\mathbb{R}_2$ defined on the basis S and S_1 respectively. $\mathbb{R}_1$ will be applied first to the vector $\boldsymbol{r}$ in S and after it will be applied $\mathbb{R}_2$ to the vector $\boldsymbol{r}_1$ in S_1. Fig. 1.9 shows these rotations.

$$S \xrightarrow{\mathbb{R}_1} S_1 \xrightarrow{\mathbb{R}_2} S_2$$

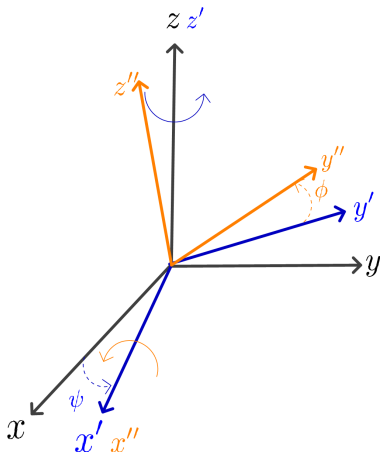


Figure 1.9: Rotations in different frameworks.

The resulting rotation matrix will be:

$$\mathbb{R} = \mathbb{R}_1\mathbb{R}_2 \qquad \text{such that} \qquad \underset{S}{\boldsymbol{r}_2} = \mathbb{R}\underset{S}{\boldsymbol{r}}$$

In other words, when the second rotation is carried out in the new basis vectors $(\boldsymbol{i}_1, \boldsymbol{j}_1, \boldsymbol{k}_1)$, the resulting rotation is obtained from a post-multiplication. In the other side, when the rotations are in the same coordinated system, the resulting rotation is obtained from a pre-multiplication.

1.3.4.3 Transformation of Coordinates Caused by a Rotation Matrix

A rotation matrix generates a new coordinated system due to rotation of a set of basis vectors $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$ as it is shown in Fig. 1.10.

$$S \xrightarrow{\mathbb{R}} S_1$$

The new basis vectors result from the rotation of the original basis vectors:

$$(\boldsymbol{i}_1, \boldsymbol{j}_1, \boldsymbol{k}_1)^T = \mathbb{R}(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})^T$$

Any vector $\mathbf{r}$ expressed in system S_1 can be expressed in the system S by doing:

$$\begin{pmatrix} coords\ of \\ \boldsymbol{r} \\ in\ S \end{pmatrix} = R \begin{pmatrix} coords\ of \\ \boldsymbol{r} \\ in\ S_1 \end{pmatrix}$$

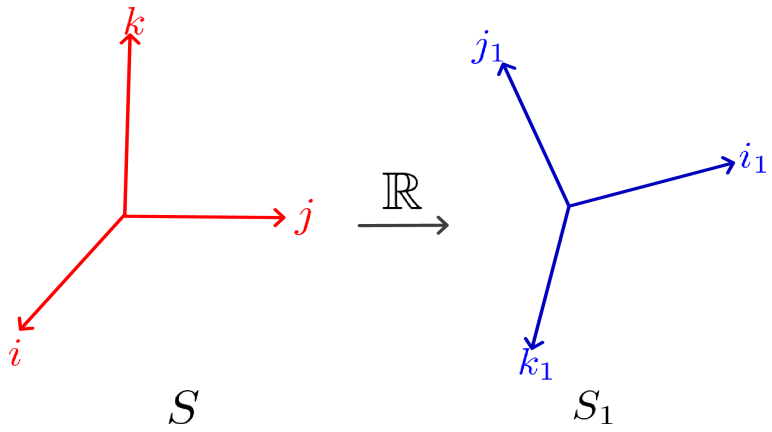


Figure 1.10: System S1 is generated by the rotation matrix ℝ.

1.3.5 Euler Angles

The Euler angles are the angles of three rotations about the axes x, y or z. There are several ways to chose the rotations, for example, $(x-y'-x'')$,$(z-x'-z'')$, $(x-y'-z'')$ and so on. Here it will be chosen the rotation $(z-y'-x'')$. The angles of this rotation about three different axes are called Cardan angles or Tait-Bryan angles.

1.3.5.1 Cardan Angles

The rotation around the axis z is named yaw angle, the rotation around y is named pitch angle and the rotation about x is the roll angle. Fig. 1.11 describes these three rotations,

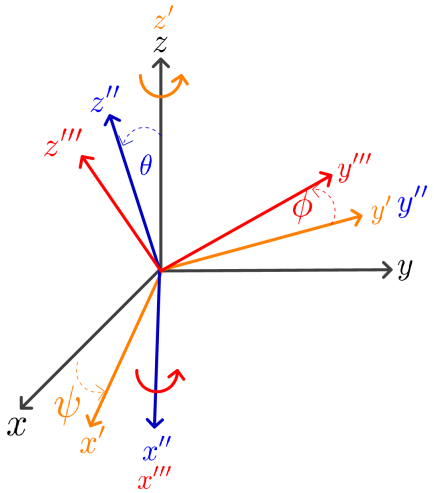


Figure 1.11: Euler angles in the zyx convention or Cardan angles.

The first rotation is performed around z axis and it originates the system S_1. The second rotation is done around y' axis and it originates the system S_2. Finally, The third one is

carried out around x'' axis and it originates the system S_3. The resulting rotation matrix is given by:

$$\mathbb{R} = \begin{pmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{pmatrix}$$

In addition, any vector expressed in the last frame S_3 can be expressed in the first frame S by means of this expression of $\mathbb{R}$.

1.3.6 Derivative of a Rotation Matrix

The derivative of a rotation is needed to know the evolution of the rotation at every time. The variation of the rotation matrix can be represented as a product of two rotations, that is:

$$\mathbb{R}(t+\delta t) = \mathbb{R}(t)\mathbb{R}(\delta t)$$

where $R(\delta t)$ represents a rotation around the new frame determined by the rotation matrix $\mathbb{R}(t)$. Besides, the definition of velocity and the Euler theorem allow to find a relationship with the skew symmetric matrix.

$$\boldsymbol{v} = \lim_{\delta t \to 0} \frac{\boldsymbol{r}(t+\delta t) - \boldsymbol{r}(t)}{\delta t} = \boldsymbol{\omega} \times \boldsymbol{r} = \mathbb{W}\boldsymbol{r}$$

If $\boldsymbol{r}(t+\delta t)$ is a consequence of a rotation, it can be written as:

$$\boldsymbol{r}(t+\delta t) = \mathbb{R}(\delta t)\boldsymbol{r}(t)$$

Next,substituting this into the velocity equation:

$$\lim_{\delta t \to 0} \frac{\mathbb{R}(\delta t)\boldsymbol{r}(t) - \boldsymbol{r}(t)}{\delta t} = \lim_{\delta t \to 0} \frac{(\mathbb{R}(\delta t) - I_3)\boldsymbol{r}(t)}{\delta t}$$

$$\boldsymbol{v} = \left(\lim_{\delta t \to 0} \frac{\mathbb{R}(\delta t) - I_3}{\delta t} \right) \boldsymbol{r}(t)$$

where I_3 is the identity matrix. Consequently,

$$\lim_{\delta t \to 0} \frac{\mathbb{R}(\delta t) - I_3}{\delta t} = W \qquad \forall \boldsymbol{r} \neq 0$$

Using the derivative definition of a rotation matrix, it yields:

$$\frac{d\mathbb{R}}{dt} = \lim_{\delta t \to 0} \frac{\mathbb{R}(t+\delta t) - \mathbb{R}(t)}{\delta t}$$

$$\frac{d\mathbb{R}}{dt} = \lim_{\delta t \to 0} \frac{\mathbb{R}(t)\mathbb{R}(\delta t) - \mathbb{R}(t)}{\delta t} = \lim_{\delta t \to 0} \frac{\mathbb{R}(t)(\mathbb{R}(\delta t) - I_3)}{\delta t}$$

$$\frac{d\mathbb{R}}{dt} = \mathbb{R}(t) \lim_{\delta t \to 0} \frac{\mathbb{R}(\delta t) - I_3}{\delta t} = \mathbb{R}(t)\mathbb{W}$$

Finally,

$$\frac{d\mathbb{R}}{dt} = \mathbb{R}(t)\mathbb{W}$$

where $\mathbb{W}$ is the skew symmetric matrix of $\boldsymbol{\omega}$ expressed in the frame generated by $\mathbb{R}(t)$.

1.3.7 Derivative of a Vector Expressed in a Rotating Frame

Let us have two coordinated systems, the second one generated from a rotation of the first one.

$$S \xrightarrow{R} S_R$$

In order to know the time derivative relative to system S of a vector r with coordinates in the system S_R, it is necessary to express that vector in coordinates relative to S.

$$\frac{d}{dt}\left(\begin{matrix}\boldsymbol{r}\\ s\end{matrix}\right) = \frac{d}{dt}\left(\mathbb{R}\boldsymbol{r}\right) = \frac{d\mathbb{R}}{dt}r + \mathbb{R}\frac{d\boldsymbol{r}}{dt} = \mathbb{R}(t)\mathbb{W}\boldsymbol{r} + \mathbb{R}\frac{d\boldsymbol{r}}{dt}$$

$$\frac{d}{dt}\left(\begin{matrix}\boldsymbol{r}\\ s\end{matrix}\right) = \mathbb{R}\left(\mathbb{W}\boldsymbol{r} + \frac{d\boldsymbol{r}}{dt}\right)$$

1.3.8 Lift and Drag Torques

The airflow striking the helices generates two forces, one called lift and another named drag. The airflow around the helices is shown in the Fig. 1.12

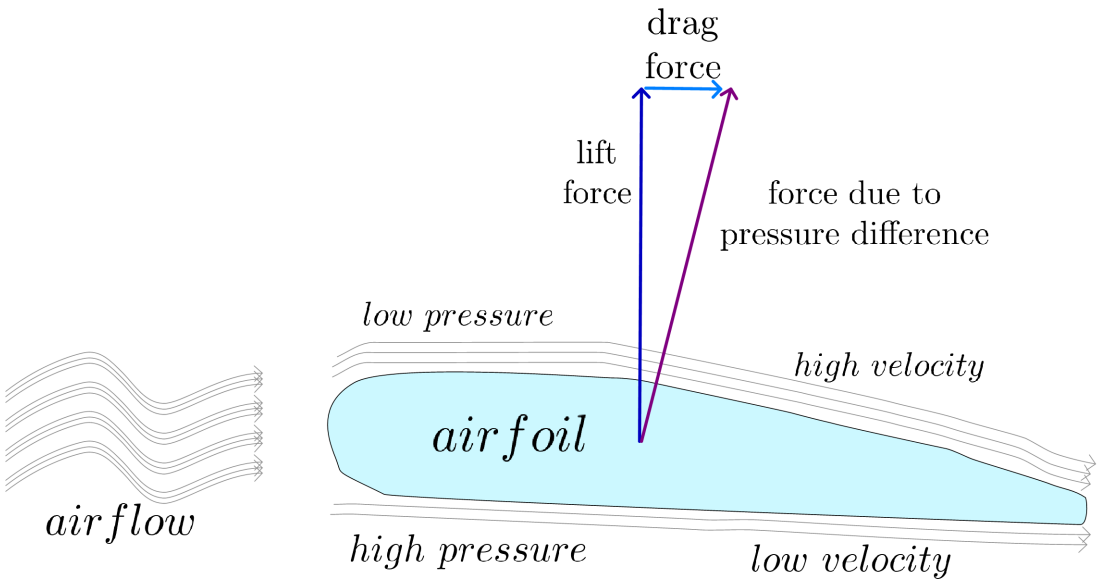


Figure 1.12: Diagram of airflow striking a turning propeller.

The low pressure is due to a high velocity and the high pressure is due to low velocity of airflow. This difference generates a perpendicular force to the flow. The flow undergoes a change in direction called upward and this change causes the apparition of drag and lift forces, as shown in Fig. 1.13:

The lift and drag forces could be assumed to be proportional to square of helices velocity.

$$\begin{aligned} f_{lift} &= k_{lift}w^2 \\ f_{drag} &= k_{drag}w^2 \end{aligned}$$

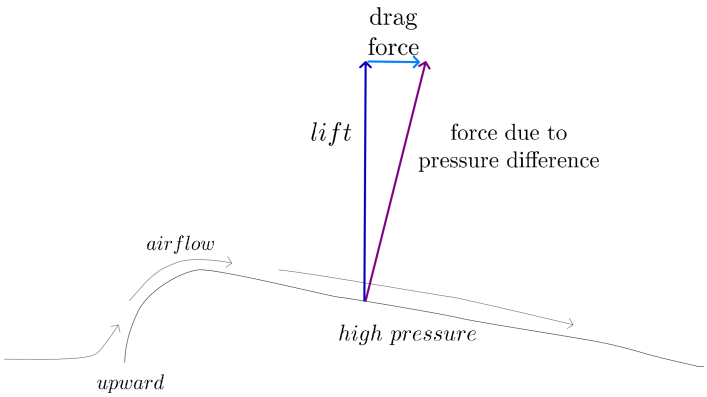


Figure 1.13: Lift and drag force generated when a propeller is turning.

1.4 Summary

In this chapter, we make a revision of the present prototype acting as a hybrid aerial and terrestrial vehicle. One of them is a VTOL that uses its wings to walk but the use of another actuator for this task. The other prototype use a quadrotor with a sort of cage, this idea let the vehicle to move fast than using supports as feet. From this conceptions, we proposed a new one, keeping fast displacement on ground, better air flow and minimal contact points with ground and as consequence the increasing of operation time. The other subsections summarized the fundamental mathematical basis of a rigid body in rotation so that next chapter focusing in the modeling of this prototype be more readable.

Chapter 2

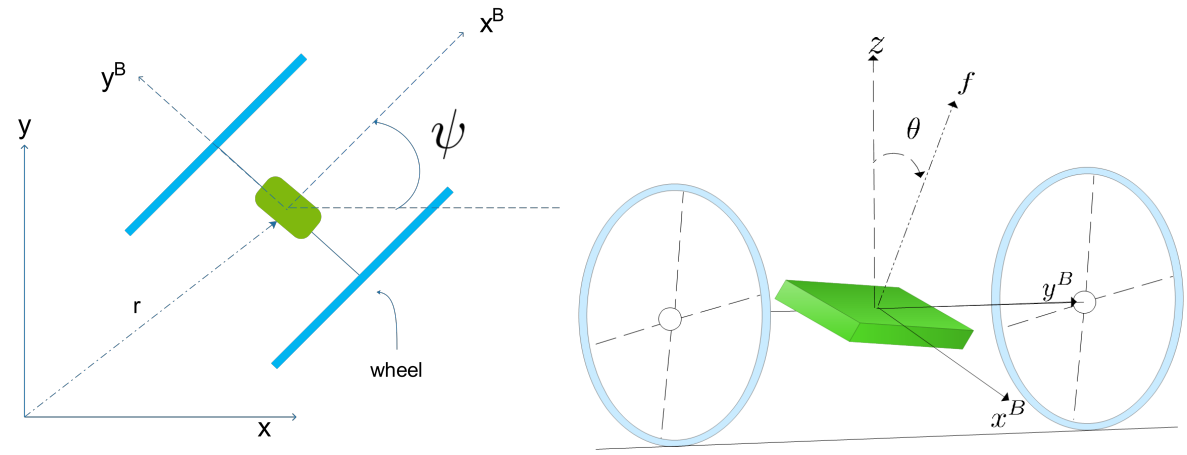
Modeling

Contents

This section focuses on the model of the proposed vehicle prototype. This vehicle can behave as a cart when it is in ground operation or as a quacopter vehicle when it is flying. There are two passive wheels attached to an axis that goes through the whole vehicle. It has also four motors with propellers that generate the force for moving the cart and, in this mode, this force is bounded in order to avoid the take-off. The cart can turn around itself, move forward or backward because it uses the thrust generated by propellers and the pitch angle for displacing and yaw angle for controlling the orientation of this movement. In aerial mode, this vehicle uses the thrust and its orientation for moving around like a quadrotor. In this mode the thrust is bigger than that used in the ground mode and this leads to a more significant energy consumption. These characteristics make suitable the vehicle for different scenarios.

2.1 Model of a ground vehicle

Several works described their models about a wheeled vehicle with their own particularities [7–11], and they cover certain topics about these kind of non-holonomic robots. Taking this works as a previous reference, the model of our vehicle acting as a cart with two passive wheels is obtained using the Newton-Euler approach. The mathematical model can be obtained from Fig. 2.1. A top and lateral view can be seen in Figs. 2.1a and 2.1b. The yaw angle ψ controls the direction of the movement and the pitch angle θ regulates its speed. Here, it is assumed the wheels are always in contact with the floor. The floor is considered perfectly flat what leads to a roll angle $\phi = 0$. The framework attached to the vehicle is denoted with a superscript letter B and the inertia framework with a superscript I.



(a) (b)

Figure 2.1: These figures describe a cart with two passive wheels. The Fig. (a) is a top-view and Fig. (b) shows a lateral perspective.

The magnitude of the main thrust f must satisfy the following constraint in order to avoid the take-off.

$$f\cos\theta \leq mg \tag{2.1}$$

where m indicates the vehicle mass and g means the gravitational acceleration. Applying the Newton second law, the model of the cart is given by

$$m\ddot{\boldsymbol{r}} = \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix} f\sin\theta \tag{2.2}$$

where $\ddot{\boldsymbol{r}}$ means the acceleration of the mass center of the cart over the xy plane. The friction force due to the ground is neglected as well as the friction in the rotation axis of the wheels. From Eqn. (2.2), the variables that can be regulated in order to place the vehicle in a desired position are f, θ, ψ. Therefore, it is necessary to obtain the orientation model in order to design an algorithm that regulates these variables for controlling the vehicle position. Under the supposition the mass center and the geometric center coincide, the theorem of Resal gives

$$\dot{\boldsymbol{L}}^{\boldsymbol{I}} = \boldsymbol{\tau}^{\boldsymbol{I}} \tag{2.3}$$

$\boldsymbol{\tau}^{\boldsymbol{I}}$ is the torques generated by helices and expressed in the inertial framework. $\boldsymbol{L}^{\boldsymbol{I}}$ defines the angular momentum expressed in the same framework than $\boldsymbol{\tau}^{\boldsymbol{I}}$. $\boldsymbol{L}^{\boldsymbol{I}}$ and $\boldsymbol{\tau}^{\boldsymbol{I}}$ can be computed from their respective expressions in the body frame, this leads to

$$\frac{d}{dt}\left(\mathbb{R}(\mathbb{I}\boldsymbol{\omega})\right) = \mathbb{R}\boldsymbol{\tau}^{\boldsymbol{B}} + \mathbb{R}\boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}} \tag{2.4}$$

The matrix $\mathbb{I}$ represents the inertia of the cart. $\boldsymbol{\omega}$ stands for the angular velocity in the body frame and $\mathbb{R}$ means the rotation matrix of the cart. $\boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}}$ indicates the torque caused by the reaction forces of the ground. Now, developing the previous expression, it yields

$$\dot{\mathbb{R}}(\mathbb{I}\boldsymbol{\omega}) + \mathbb{R}\mathbb{I}\dot{\boldsymbol{\omega}} = \mathbb{R}(\boldsymbol{\tau}^{\boldsymbol{B}} + \boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}}) \tag{2.5}$$

$$\mathbb{R}[\boldsymbol{\omega}]^{\times}(\mathbb{I}\boldsymbol{\omega}) + \mathbb{R}\mathbb{I}\dot{\boldsymbol{\omega}} = \mathbb{R}(\boldsymbol{\tau}^{\boldsymbol{B}} + \boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}}) \tag{2.6}$$

with $[\boldsymbol{\omega}]^\times$ represents the skew symmetric matrix of $\boldsymbol{\omega}$. The matrix $[\boldsymbol{\omega}]^\times$ is defined as

$$[\boldsymbol{\omega}]^\times = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & \omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \tag{2.7}$$

Therefore, from Eqn. (2.6) it follows the attitude dynamics is expressed as

$$\mathbb{I}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} - [\boldsymbol{\omega}]^\times \mathbb{I}\boldsymbol{\omega} + \boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}} \tag{2.8}$$

From the geometry of the vehicle, it can be deduced that the torque generated by the ground $\boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}}$ belongs to the $x^B z^B$ plane. From Fig. 2.2, the $\boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}}$ can be expressed as

$$\boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}} = \begin{pmatrix} \cos\theta \\ 0 \\ \sin\theta \end{pmatrix} |\boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}}| \tag{2.9}$$

or alternatively,

$$\boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}} = \mathbf{R}_y(-\theta) \begin{pmatrix} |\boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}}| \\ 0 \\ 0 \end{pmatrix} \tag{2.10}$$

where $\mathbf{R}_y(-\theta)$ stands for a rotation about y of an angle $-\theta$.

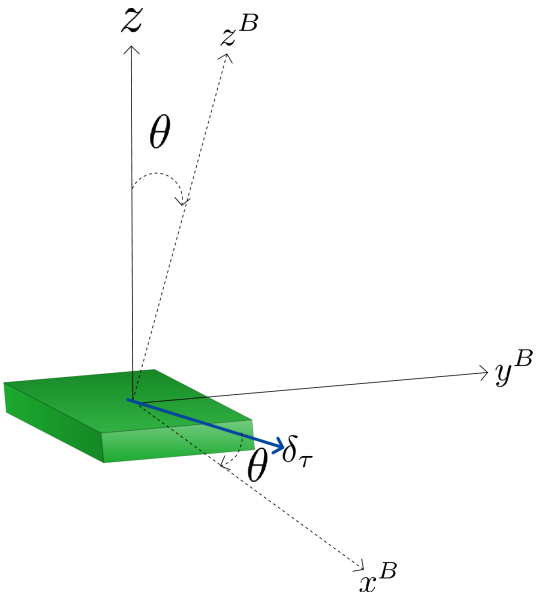


Figure 2.2: Torque generated by the reaction forces of the ground.

Now, it is necessary to find the relationship between the evolution of the Euler angles and the angular velocity of the vehicle. The Fig. 2.3 shows the frameworks generated by the rotations around ψ, θ and ϕ. The variables ψ, θ and ϕ mean respectively the yaw, pitch and roll angles. The last framework generated by these rotations is the body frame.

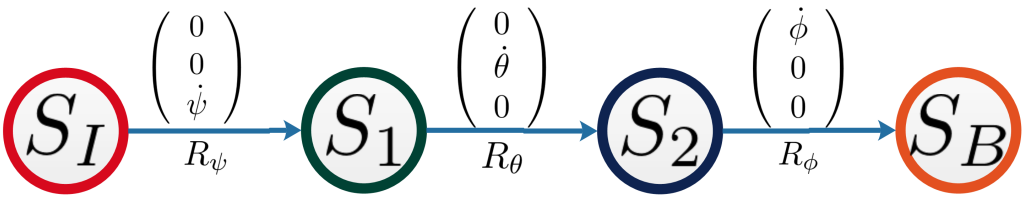


Figure 2.3: Rotation Scheme. S_i represents the frameworks generated by the rotations. $\mathbb{R}_\psi$ generates the system S_1, $\mathbb{R}_\theta$ generates the system S_2 and $\mathbb{R}_\phi$ generates the system S_B.

From this, the angular velocity can be expressed as the result of these rotations

$$\boldsymbol{\omega} = \mathbb{R}_\phi^T \mathbb{R}_\theta^T \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} + \mathbb{R}_\phi^T \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} \tag{2.11}$$

where all the velocities are expressed in the body system. $\mathbb{R}_\phi$ stands for a rotation of angle ϕ around the x axis. $\mathbb{R}_\theta$ indicates the rotation of angle θ around the y axis.

$$\mathbb{R}_\phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \qquad \mathbb{R}_\theta = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.12}$$

and because of the supposition the ground is completely flat and horizontal, the matrix $\mathbb{R}_\phi$ reduces to the identity $\mathbb{I}$. Therefore, the expression linking the angular velocity and the derivative of Euler angles becomes

$$\boldsymbol{\omega} = \begin{pmatrix} -\sin\theta & 0 & 1 \\ 0 & 1 & 0 \\ \cos\theta & 0 & 0 \end{pmatrix} \dot{\boldsymbol{\eta}} \tag{2.13}$$

where $\boldsymbol{\eta} = [\psi,\ \theta,\ \phi]^T$ stands for the Euler angles in the zyx convention. Eqn. (2.13) can be rearranged as

$$\begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \sec\theta \\ 0 & 1 & 0 \\ 1 & 0 & \tan\theta \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{2.14}$$

and from Eqn. (2.14) and with the supposition $\dot{\phi} = 0$, it is gathered that

$$\omega_x = -\omega_z \tan\theta \tag{2.15}$$

Finally, the model of the cart can be established by

$$\begin{aligned} m\ddot{x} &= f\cos\psi\sin\theta \\ m\ddot{y} &= f\sin\psi\sin\theta \\ \dot{\theta} &= \omega_y^B \\ \dot{\psi} &= \omega_z^B\ \sec\theta \\ \mathbb{I}\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau} - [\boldsymbol{\omega}]^\times \mathbb{I}\boldsymbol{\omega} + \mathbb{R}_y(-\theta)\ [|\boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{B}}|\ 0\ 0]^T \end{aligned} \tag{2.16}$$

2.2 Model of a quadrotor

The quadrotor model has well known and it has been the object of a wide range of studies [12–14]. Some model are deducted from energetic methods, others use the Euler-Newton approach and more recent works include quaternions for modeling the attitude part of the system [15–17]. Here, the Euler-Newton approach is preferred to model our prototype because it gives a physical understanding of its dynamics. It is supposed the location of the geometric center and the mass one to be the same. The force acting over the drone will be its weight and the forces generated by the helices rotation. In this work we consider that propellers forces are on the z axis of the body framework, therefore the main thrust stays on the same axis. Fig. 2.4 illustrates a description of forces and torques acting on the drone. In order to have a simple model, some dynamics are not considered, as the flapping and flexibility of helices, ground effect when the fly is at low height, possible vibrations of its structure when flying, response time of motors, among others. These neglected dynamics could be counteracted by some control techniques and having a simple model let us to have a better overview of the system.

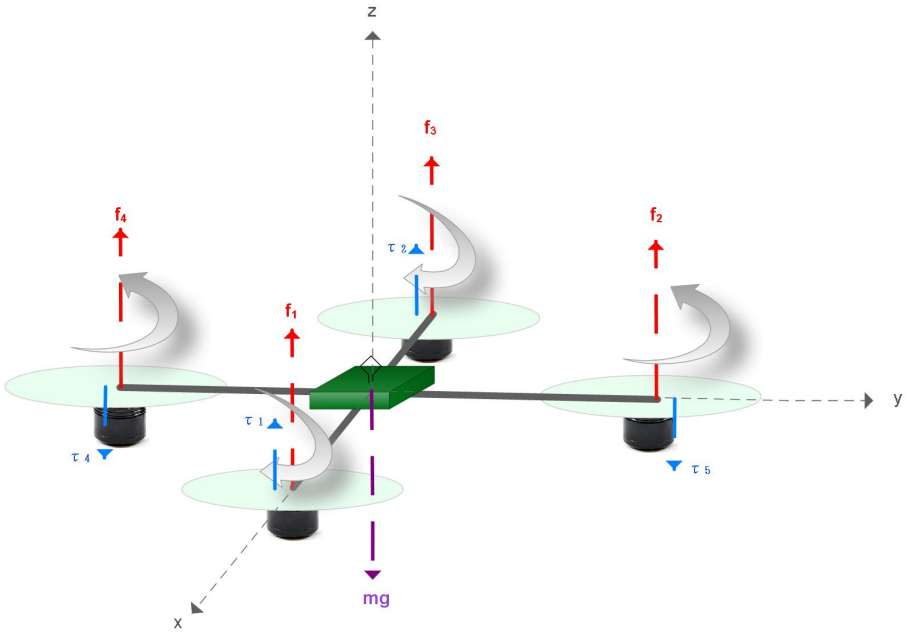


Figure 2.4: Forces and torques diagram of a quadrotor.

Using the second law of Newton, the position dynamics can be written as

$$m\ddot{\boldsymbol{r}} = \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} \tag{2.17}$$

where $\mathbb{R}$ represents the rotation matrix, $\boldsymbol{F}$ means the resultant force generated by the helices and $\boldsymbol{F_g}$ indicates the weight. The matrix $\mathbb{R}$ gives the orientation of the vehicle. From Figure 2.3, the expression for $\mathbb{R}$ can be deduced as

$$\mathbb{R} = \mathbb{R}_\psi \mathbb{R}_\theta \mathbb{R}_\phi$$

which leads to

$$\mathbb{R} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix}$$

then,

$$\mathbb{R} = \begin{pmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{pmatrix} \tag{2.18}$$

Under the supposition the thrust generated by propellers is parallel to z axis, then,

$$\boldsymbol{F} = \begin{pmatrix} 0 \\ 0 \\ f \end{pmatrix}$$

where f defines the magnitude of the thrust. Therefore, the position dynamics for the quadrotor becomes

$$m \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ \cos\theta\cos\phi \end{pmatrix} f + \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} \tag{2.19}$$

and its orientation dynamics can be obtained by the Resal theorem,

$$\frac{d}{dt}\left(\mathbb{R}(\mathbb{I}\boldsymbol{\omega})\right) = \mathbb{R}\boldsymbol{\tau}$$

which leads to

$$\mathbb{I}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} - [\boldsymbol{\omega}]^{\times}\mathbb{I}\boldsymbol{\omega} \tag{2.20}$$

where $\mathbb{I}$ represents the inertia matrix, $\boldsymbol{\omega}$ means the angular velocity of the quadrotor and $\boldsymbol{\tau}$ indicates the torques in the body frame. Now, it is necessary to determine the relationship between the zyx Euler angles and the angular velocity. From (2.3), the angular rate is expressed in terms of the successive rotations about the axes z, y and x. Thus,

$$\boldsymbol{\omega} = \mathbb{R}_{\phi}^{T}\mathbb{R}_{\theta}^{T} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} + \mathbb{R}_{\phi}^{T} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} \tag{2.21}$$

then,

$$\boldsymbol{\omega} = \begin{pmatrix} -\sin\theta & 0 & 1 \\ \sin\phi\cos\theta & \cos\phi & 0 \\ \cos\phi\cos\theta & -\sin\phi & 0 \end{pmatrix} \dot{\boldsymbol{\eta}} \tag{2.22}$$

or conversely,

$$\begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \\ 0 & \cos\phi & -\sin\phi \\ 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{2.23}$$

Now, the model of a quadrotor can be established by

$$\begin{aligned} m\ddot{\boldsymbol{r}} &= \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} \\ \dot{\boldsymbol{\eta}} &= \mathbb{B}(\boldsymbol{\eta})\ \boldsymbol{\omega} \\ \mathbb{I}\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau} - [\boldsymbol{\omega}]^{\times}\mathbb{I}\boldsymbol{\omega} \end{aligned} \tag{2.24}$$

where the matrix $\mathbb{B}(\boldsymbol{\eta})$ is deduced from (2.23).

2.3 Model of the hybrid vehicle

Taking the two models developed in the previous sections for both the cart and the quadrotor, the complete model for this terrestrial and aerial vehicle, shown in Figure 2.5, could be written in the form of (2.25),

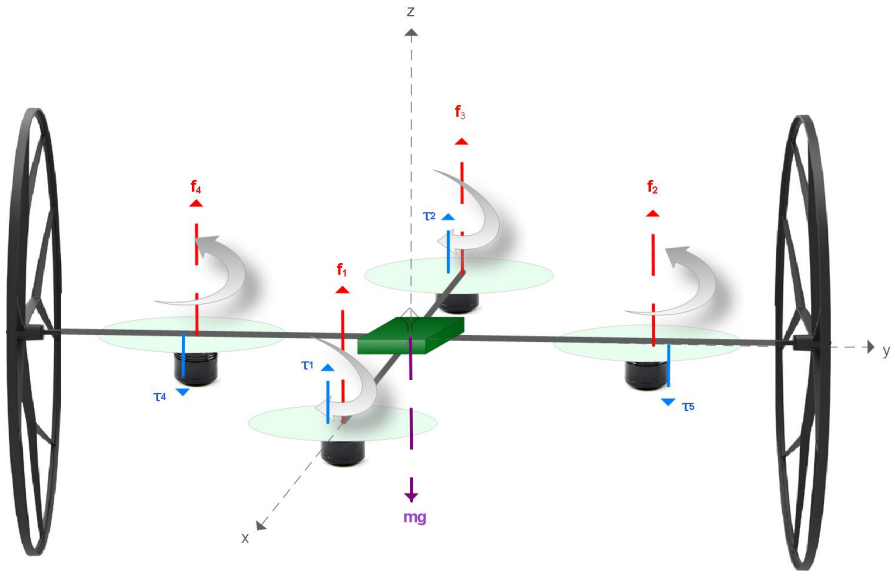


Figure 2.5: Hybrid Vehicle

$$\begin{aligned} m\ddot{\boldsymbol{r}} &= \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} + \boldsymbol{\delta_u} \\ \dot{\boldsymbol{\eta}} &= \mathbb{B}(\boldsymbol{\eta})\boldsymbol{\omega} \\ \mathbb{J}\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau} - [\boldsymbol{\omega}]^{\times}\mathbb{J}\boldsymbol{\omega} + \boldsymbol{\delta_\tau} \end{aligned} \tag{2.25}$$

where the superscript are been removed in order to have a clear reading. The matrix $\mathbb{B}(\boldsymbol{\eta})$ can be deduced from (2.23)

2.3.1 Aerial Mode

In this mode, the vehicle behaves as a quadrotor with two wheels. The term $\boldsymbol{\delta_u}$ is considered zero.

$$\boldsymbol{\delta_u} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.26}$$

This means that when the vehicle is flying, it moves only under the action of its weight $\boldsymbol{F_g}$ and of the thrust $\boldsymbol{F}$ generated by the helices. In a similar way, we consider the term $\boldsymbol{\delta_\tau}$ as

$$\boldsymbol{\delta_\tau} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.27}$$

This means that the vehicle is supposed to operate only under the torques produced by its propellers. Consequently, the simplified model for flying mode becomes

$$\begin{aligned} m\ddot{\boldsymbol{r}} &= \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} \\ \dot{\boldsymbol{\eta}} &= \mathbb{B}(\boldsymbol{\eta})\boldsymbol{\omega} \\ \mathbb{J}\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau} - [\boldsymbol{\omega}]^{\times}\mathbb{J}\boldsymbol{\omega} \end{aligned} \tag{2.28}$$

Remind the convention *yaw-pitch-roll* or zyx is used for the Euler angles.

2.3.2 Ground Mode

In this mode, the vehicle behaves as cart. The reaction forces and torques due to the contact to the ground appear. The terms $\boldsymbol{\delta_u}$ and $\boldsymbol{\delta_\tau}$ will be non-zero. In order to make easier the reading, three frameworks are determined. The inertial framework is identified by the letter I, the body framework by the B and a new framework by the G. This last one is parallel to the ground and is attached to the body. Fig. 2.6 could help to identify the frameworks.

We suppose the vehicle moves on a flat surface and is always in touch with the ground. The position model is in the framework I and the orientation part in the framework B. The term $\boldsymbol{\delta_u} = \boldsymbol{\delta_u^I}$ is considered as

$$\boldsymbol{\delta_u^I} = \begin{pmatrix} 0 \\ 0 \\ \delta_{uz}^I \end{pmatrix} \tag{2.29}$$

This represents the ground reaction over the cart. Like the z-velocity is zero in cart mode, then

$$\delta_{uz}^I = mg - (\mathbb{R}_{I,B}\boldsymbol{F})_z \geq 0 \tag{2.30}$$

and it is always non negative. $(\mathbb{R}_{I,B}\boldsymbol{F})_z$ stands for the z-component of the force generated by the propellers. $\mathbb{R}_{I,B} = \mathbb{R}$ is the rotation matrix of the body with respect to inertial frame. The vehicle height will be constant and it may be different from zero. When the vehicle is over ground, it appears a reaction torque in x^G axis which prevents to turn around it.

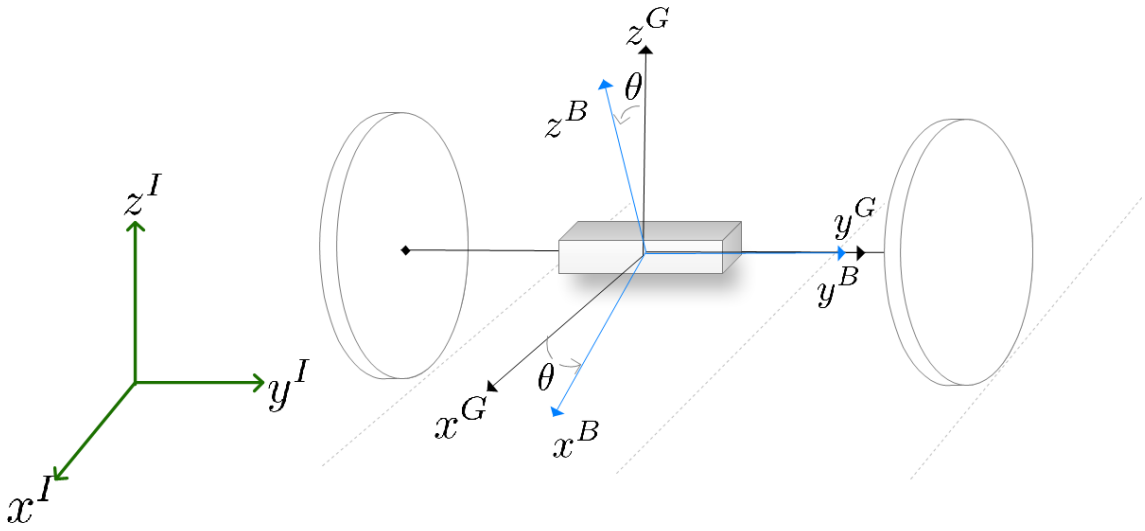


Figure 2.6: Frameworks I, B and G

Consequently, the cart will stay attached to the surface. The torque produced by friction over axes y^G and z^G are neglected. Then, the torque in the G-framework is determined by

$$\boldsymbol{\delta}_{\boldsymbol{\tau}}^G = \begin{pmatrix} \delta_{\tau_x}^G \\ 0 \\ 0 \end{pmatrix} \tag{2.31}$$

The link between the G and B frameworks is made by the matrix $\mathbb{R}_{G,B}$

$$\mathbb{R}_{G,B} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.32}$$

Now, the orientation model can be written as

$$\mathbb{J}\dot{\boldsymbol{\omega}}^B = \boldsymbol{\tau}^B - [\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B + \mathbb{R}_{G,B}^T \boldsymbol{\delta}_{\boldsymbol{\tau}}^G \tag{2.33}$$

or alternatively,

$$\mathbb{R}_{G,B}\mathbb{J}\dot{\boldsymbol{\omega}}^B = \boldsymbol{\tau}^G - \mathbb{R}_{G,B}([\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B) + \boldsymbol{\delta}_{\boldsymbol{\tau}}^G \tag{2.34}$$

From (2.6), it is deduced that

$$\boldsymbol{\omega}^G = \mathbb{R}_{G,B}\boldsymbol{\omega}^B \quad \longleftrightarrow \quad \boldsymbol{\omega}^B = \mathbb{R}_{G,B}^T \boldsymbol{\omega}^G \tag{2.35}$$

and consequently,

$$\dot{\boldsymbol{\omega}}^B = \dot{\mathbb{R}}_{G,B}^T \boldsymbol{\omega}^G + \mathbb{R}_{G,B}^T \dot{\boldsymbol{\omega}}^G \tag{2.36}$$

Then, developing the terms of previous equations in order to obtain an expression for $\delta_{\tau_x}^G$, it follows that

$$\dot{\mathbb{R}}_{G,B}^T \boldsymbol{\omega}^G = \begin{pmatrix} -s\theta & 0 & -c\theta \\ 0 & 0 & 0 \\ c\theta & 0 & -s\theta \end{pmatrix} \dot{\theta} \begin{pmatrix} 0 \\ \omega_y^G \\ \omega_z^G \end{pmatrix} = \begin{pmatrix} -c\theta\omega_z^G \\ 0 \\ -s\theta\omega_z^G \end{pmatrix} \dot{\theta} \tag{2.37}$$

$$\mathbb{R}^T_{G,B}\dot{\boldsymbol{\omega}}^G = \begin{pmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{pmatrix} \begin{pmatrix} 0 \\ \dot{\omega}^G_y \\ \dot{\omega}^G_z \end{pmatrix} = \begin{pmatrix} -s\theta\dot{\omega}^G_z \\ \dot{\omega}^G_y \\ c\theta\dot{\omega}^G_z \end{pmatrix} \tag{2.38}$$

The inertia matrix is considered as diagonal $\mathbb{J} = \text{diag}(J_{xx}, J_{yy}, J_{zz})$. Then

$$\mathbb{R}_{G,B}\mathbb{J} = \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix} \begin{pmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{pmatrix} = \begin{pmatrix} J_{xx}c\theta & 0 & J_{zz}s\theta \\ 0 & J_{yy} & 0 \\ -J_{xx}s\theta & 0 & J_{zz}c\theta \end{pmatrix} \tag{2.39}$$

Now, developing $R_{G,B}\mathbb{J}(\dot{\boldsymbol{\omega}}^B) = R_{G,B}\mathbb{J}(\dot{R}^T_{G,B}\boldsymbol{\omega}^G + R^T_{G,B}\dot{\boldsymbol{\omega}}^G)$, it results

$$\begin{aligned} \mathbb{R}_{G,B}\mathbb{J}\dot{\mathbb{R}}^T_{G,B}\boldsymbol{\omega}^G &= \begin{pmatrix} J_{xx}c\theta & 0 & J_{zz}s\theta \\ 0 & J_{yy} & 0 \\ -J_{xx}s\theta & 0 & J_{zz}c\theta \end{pmatrix} \begin{pmatrix} -c\theta\omega^G_z \\ 0 \\ -s\theta\omega^G_z \end{pmatrix} \dot{\theta} \\ &= \begin{pmatrix} -J_{xx}\omega^G_z c^2\theta - J_{zz}\omega^G_z s^2\theta \\ 0 \\ J_{xx}\omega^G_z s\theta c\theta - J_{zz}\omega^G_z s\theta c\theta \end{pmatrix} \dot{\theta} \end{aligned} \tag{2.40}$$

$$\begin{aligned} \mathbb{R}_{G,B}\mathbb{J}\mathbb{R}^T_{G,B}\dot{\boldsymbol{\omega}}^G &= \begin{pmatrix} J_{xx}c\theta & 0 & J_{zz}s\theta \\ 0 & J_{yy} & 0 \\ -J_{xx}s\theta & 0 & J_{zz}c\theta \end{pmatrix} \begin{pmatrix} -s\theta\dot{\omega}^G_z \\ \dot{\omega}^G_y \\ c\theta\dot{\omega}^G_z \end{pmatrix} \\ &= \begin{pmatrix} -J_{xx}\dot{\omega}^G_z s\theta c\theta + J_z z\dot{\omega}^G_z s\theta c\theta \\ J_{yy}\dot{\omega}^G_y \\ J_{xx}\dot{\omega}^G_z s^2\theta + J_{zz}\dot{\omega}^G_z c^2\theta \end{pmatrix} \end{aligned} \tag{2.41}$$

similarly, computing the expression for $\mathbb{R}_{G,B}([\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B)$.

$$[\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B = \begin{pmatrix} \omega^B_x \\ \omega^B_y \\ \omega^B_z \end{pmatrix}^\times \begin{pmatrix} J_{xx}\omega^B_x \\ J_{yy}\omega^B_y \\ J_{zz}\omega^B_z \end{pmatrix} = \begin{pmatrix} \omega^B_y\omega^B_z(J_{zz} - J_{yy}) \\ \omega^B_x\omega^B_z(J_{xx} - J_{zz}) \\ \omega^B_x\omega^B_y(J_{yy} - J_{xx}) \end{pmatrix} \tag{2.42}$$

and with

$$\begin{aligned} \boldsymbol{\omega}^B = \mathbb{R}^T_{G,B}\boldsymbol{\omega}^G &= \begin{pmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{pmatrix} \begin{pmatrix} \omega^G_x \\ \omega^G_y \\ \omega^G_z \end{pmatrix} \\ \boldsymbol{\omega}^B &= \begin{pmatrix} -s\theta\omega^G_z \\ \omega^G_y \\ c\theta\omega^G_z \end{pmatrix} \end{aligned} \tag{2.43}$$

Therefore, the expression (2.42) can be rewritten in terms of $\boldsymbol{\omega}^G$ as

$$[\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B = \begin{pmatrix} \omega^G_y\omega^G_z c\theta\ (J_{zz} - J_{yy}) \\ -(\omega^G_z)^2 s\theta c\theta\ (J_{xx} - J_{zz}) \\ \omega^G_y\omega^G_z s\theta\ (J_{xx} - J_{zz}) \end{pmatrix} \tag{2.44}$$

Finally, $\mathbb{R}_{G,B}([\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B)$ becomes

$$\begin{aligned}\mathbb{R}_{G,B}([\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B) &= \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix} \begin{pmatrix} \omega_y^G \omega_z^G c\theta\ (J_{zz} - J_{yy}) \\ -(\omega_z^G)^2 s\theta c\theta\ (J_{xx} - J_{zz}) \\ \omega_y^G \omega_z^G s\theta\ (J_{xx} - J_{zz}) \end{pmatrix} \\ &= \begin{pmatrix} \omega_y^G\ \omega_z^G[c^2\theta(J_{zz} - J_{yy}) - s^2\theta(J_{yy} - J_{xx})] \\ -(\omega_z^G)^2 s\theta c\theta\ (J_{xx} - J_{zz}) \\ \omega_y^G \omega_z^G c\theta s\theta\ (J_{xx} - J_{zz}) \end{pmatrix}\end{aligned} \tag{2.45}$$

Now there are all the elements for finding the expression for $\delta_{\tau_x}^G$. Taking the x-component of (2.34), it results:

$$\delta_{\tau_x}^G = -(J_{xx}c^2\theta + J_{zz}s^2\theta)\omega_z^G\dot{\theta} - (J_{xx} - J_{zz})\dot{\omega}_z^G s\theta c\theta - \tau_x^G + [\mathbb{R}_{G,B}([\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B)]_x \tag{2.46}$$

From z-component of (2.34), the expression for $\dot{\omega}_z^G$ is

$$\dot{\omega}_z^G = \frac{\tau_z^G - [\mathbb{R}_{G,B}([\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B)]_z - (J_{xx} - J_{zz})\omega_z^G s\theta c\theta\dot{\theta}}{J_{xx}s^2\theta + J_{zz}c^2\theta} \tag{2.47}$$

Therefore, by replacing (2.47) into (2.46) and taking $\dot{\theta} = \omega_y^G$ and $\tau_x^G = 0$, it yields

$$\begin{aligned}\delta_{\tau_x}^G = \ & -(J_{xx}c^2\theta + J_{zz}s^2\theta)\omega_z^G\omega_y^G + [\mathbb{R}_{G,B}([\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B)]_x \\ & -(J_{xx} - J_{zz})s\theta c\theta \left(\frac{\tau_z^G - [\mathbb{R}_{G,B}([\boldsymbol{\omega}^B]^\times \mathbb{J}\boldsymbol{\omega}^B)]_z - (J_{xx} - J_{zz})\omega_z^G s\theta c\theta \omega_y^G}{J_{xx}s^2\theta + J_{zz}c^2\theta} \right)\end{aligned} \tag{2.48}$$

Additionally, $\omega_x^G = \cos\theta \cdot \omega_x^B + \sin\theta \cdot \omega_z^B = 0$. Then, when the vehicle is over ground it holds

$$\omega_x^B = -\tan\theta \cdot \omega_z^B \tag{2.49}$$

In addition, the expression $\dot{\boldsymbol{\eta}} = \mathbb{B}(\boldsymbol{\eta})\boldsymbol{\omega}^B$, with

$$\mathbb{B}(\boldsymbol{\eta}) = \begin{pmatrix} 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \\ 0 & \cos\phi & -\sin\phi \\ 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \end{pmatrix} \tag{2.50}$$

and under the supposition the cart moves over a horizontal surface, it becomes

$$\dot{\psi} = \omega_z^B \sec\theta \tag{2.51}$$

$$\dot{\theta} = \omega_y^B \tag{2.52}$$

$$\dot{\phi} = 0 \tag{2.53}$$

In summary, the model described in equation (2.25) takes the following form,

$$\begin{aligned} m\ddot{x} &= \cos\psi\sin\theta\ f \\ m\ddot{y} &= \sin\psi\sin\theta\ f \\ m\ddot{z} &= 0 \\ \dot{\psi} &= \omega_z^B \sec\theta \\ \dot{\theta} &= \omega_y^B \\ \dot{\phi} &= 0 \\ \mathbb{J}\dot{\boldsymbol{\omega}}^B &= \boldsymbol{\tau}^B - \boldsymbol{\omega}^{B\times}\mathbb{J}\boldsymbol{\omega}^B + \mathbb{R}_{G,B}{}^T\boldsymbol{\delta}_{\boldsymbol{\tau}}^G \end{aligned} \tag{2.54}$$

with the conditions

$$\begin{aligned} \omega_x^B &= -\omega_z^B \tan\theta \\ \tau_x^B &= -\tau_z^B \tan\theta \\ f &\leq mg/\cos\theta \end{aligned} \tag{2.55}$$

Fig. 2.7 provides a lateral view of the torques and forces that could lead to detach one wheel. The force $F_z = mg - f\cos\theta$ represents the total reaction force of the ground. The forces in the contact points P_1 and P_2 generates the torque $\boldsymbol{\delta_\tau^G}$. The factor $\lambda \in [0, 1]$ and l is the length between the two wheels. When the vehicle does not detach the wheels, we obtain

$$|\boldsymbol{\delta_\tau^G}| = (1 - 2\lambda)F_z \; l/2 \tag{2.56}$$

In the instant, when one wheel will get detached, it happens $\lambda = 0$ and consequently, in order

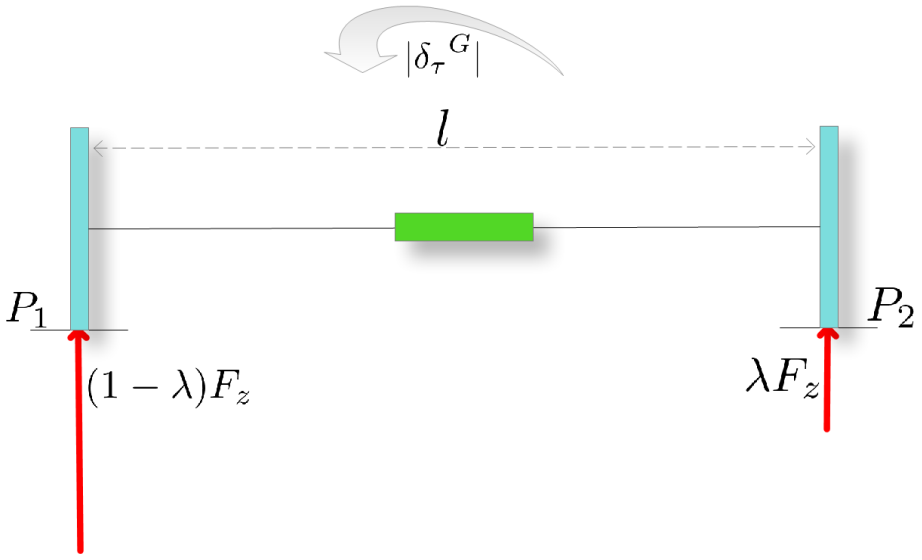


Figure 2.7: Diagram of torques and forces when vehicle could detach a wheel.

to avoid any wheel get detached, it will be required the condition

$$|\delta_{\delta_x}^G| \leq (mg - f\cos\theta)\; l/2 \tag{2.57}$$

this condition can be added to the set (2.55).

2.4 Summary

The model of the hybrid vehicle was obtained. First, the ground model was developed and the conditions it must satisfy. Then, the model for the air operation was deduced. Lastly, a model for the hybrid vehicle is proposed and it is verified this hybrid model leads to the two operation modes under their corresponding assumptions. Furthermore, the condition to avoid any of wheels get detached was computed. Based in this model, the next chapter focuses in developing control algorithm for the displacement of the vehicle in both modes.

Chapter 3

Control Algorithms

Contents

The goal of the control algorithms here is to reach the position reference by making the vehicle to hold a convenient attitude and to have a specific thrust. In literature, we can find numerous articles solving a specific challenge. Therefore, we found techniques based on saturations, sliding modes, backstepping with certain properties [18–26] that could be useful for controlling the hybrid vehicle. Other works focus on the development of algorithm to stabilize their vehicles [27–31] and other focus in doing tasks as the following of trajectories [32–35]. Also, the question of controlling a wheeled robot with non-holonomic constraints is discussed [36–40], generating smooth path in order to get to the target position. These works served as a reference for the control algorithms developed for our hybrid vehicle.

The proposed control methodology is shown in Fig. 3.1. The first step is to design a position control algorithm that allows to find a desired force. If the vehicle is under this force, then it will move to the desired position. In order to generate this force, the vehicle needs to follow a specific angular trajectory and also to generate a specific thrust. The next step is to convert or to translate this force into its corresponding desired Euler angles and thrust. In a

Figure 3.1: Control design for the hybrid vehicle

further step, this desired orientation is used in the attitude algorithm for generating a desired torque. This torque will make the vehicle follows the desired attitude.

The conversion is made using the first equation in (2.25). Fig. 3.2 shows the relationship between the desired force and its corresponding Euler angles and thrust. In the following sections will be described some strategies proposed for controlling the hybrid vehicle.

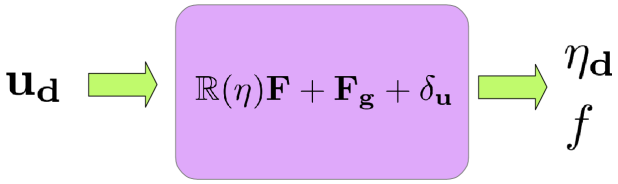


Figure 3.2: Relation between the desired force and the Euler angles and thrust. $\boldsymbol{u_d}$ is the desired control force and $\boldsymbol{\eta_d}$ and f are the desired Euler angles and thrust respectively. $\boldsymbol{F_g}$ stands for the weight and $\boldsymbol{\delta_u}$ is an external perturbations or unknown dynamics.

3.1 Nested Saturations

This technique has been used for the position control algorithm. It consists in computing the force that will make the vehicle to move with a specific velocity and once this is achieved the vehicle proceed to reach the desired position or trajectory. The saturations let to bound the force and by consequently to limit the maximum energy power consumed by the vehicle. This technique also produces smooth movements because the saturations control the reactivity of the vehicle.

3.1.1 Cart Operation Mode

The saturation control will help to limit the required thrust and by consequence the maximum desired pitch. The pitch angle regulates the speed of the vehicle over the ground and it allows the cart to go forward o backward. The yaw angle controls the direction of the movement of the vehicle. Thus, the thrust, pitch angle and yaw angle will govern the vehicle behavior. The position model showed in (2.2) can be rewritten as

$$\dot{\boldsymbol{r}} = \boldsymbol{v} \tag{3.1}$$

$$m\dot{\boldsymbol{v}} = \boldsymbol{u} \tag{3.2}$$

where v means the velocity, u represents the force and

$$\boldsymbol{r} = \begin{pmatrix} x \\ y \end{pmatrix}, \ \boldsymbol{u} = \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix} \sin\theta \cdot f \tag{3.3}$$

The desired trajectory $\boldsymbol{r_d} = [x_d, \ y_d]^T$ can be included in the model, thus the model can be stated,

$$\begin{aligned} \dot{\boldsymbol{e}}_{\boldsymbol{r}} &= \boldsymbol{p} \\ m\dot{\boldsymbol{p}} &= \boldsymbol{u} - m\ddot{\boldsymbol{r}}_{\boldsymbol{d}} \end{aligned} \tag{3.4}$$

where $\boldsymbol{e_r}$ stands for the position error and $\boldsymbol{p}$ indicates the velocity error. These are defined as

$$\begin{aligned} \boldsymbol{e}_{\boldsymbol{r}} &= \boldsymbol{r} - \boldsymbol{r}_{\boldsymbol{d}} \\ \boldsymbol{p} &= \boldsymbol{v} - \dot{\boldsymbol{r}}_{\boldsymbol{d}} \end{aligned} \tag{3.5}$$

The saturation function is defined as

$$\sigma_M(s) = \begin{cases} M & if \ s \geq M \\ s & if \ |s| < M \\ -M & if \ s \leq -M \end{cases} \tag{3.6}$$

where s is a scalar and M defines a positive constant. In the following, it will be developed the control law corresponding to the x coordinate. Considering $\boldsymbol{u}$, $\boldsymbol{p}$, $\boldsymbol{e_r}$ and $\boldsymbol{r_d}$ as vectors with two components

$$\boldsymbol{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} \ \boldsymbol{p} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} \tag{3.7}$$

$$\boldsymbol{e_r} = \begin{pmatrix} e_x \\ e_y \end{pmatrix} \ \boldsymbol{r_d} = \begin{pmatrix} r_{dx} \\ r_{dy} \end{pmatrix} \tag{3.8}$$

and proposing

$$u_x = m \ \ddot{r}_{dx} - m \ K_{1x} \ \sigma_{M_{1x}} \left(K_{px} \ p_x + K_{1x}^{-1} K_{2x} \ \sigma_{M_{2x}} \right) \tag{3.9}$$

where $M_{1x} \geq K_{1x}^{-1} K_{2x} M_{2x}$ and K_{1x}, M_{1x}, K_{px}, K_{2x} and M_{2x} are positive constant used for tuning the saturation control. Therefore, the second equation of (3.4) becomes

$$\dot{p}_x = -K_{1x} \sigma_{M_{1x}} (K_{px} p_x + K_{1x}^{-1} K_{2x} \sigma_{M_{2x}}) \tag{3.10}$$

This implies there exists a time T_1 large enough such that:

$$|K_{px} p_x| \leq K_{1x}^{-1} K_{2x} M_{2x} \quad \forall \ t \geq T_1 \tag{3.11}$$

and taking

$$M_{1x} \geq 2 K_{1x}^{-1} K_{2x} M_{2x} \tag{3.12}$$

the saturation $\sigma_{M_{1x}}$ results in

$$\sigma_{M_{1x}}(\cdot) = K_{px} p_x + K_{1x}^{-1} K_{2x} \sigma_{M_{2x}} \quad \forall \ t \geq T_1 \tag{3.13}$$

which leads to

$$\dot{p}_x = -K_{1x}K_{px}p_x - K_{2x}\sigma_{M_{2x}} \quad \forall\, t \geq T_1 \tag{3.14}$$

Now, propose

$$q_x = K_{1x}K_{px}e_x + p_x \tag{3.15}$$

and by differentiating this variable it yields

$$\dot{q_x} = -K_{2x}\sigma_{M_{2x}} \tag{3.16}$$

and by choosing $\sigma_{M_{2x}} = \sigma_{M_{2x}}(K_{qx}q_x)$ it results

$$\dot{q_x} = -K_{2x}\sigma_{M_{2x}}(K_{qx}q_x) \tag{3.17}$$

with K_{qx} as a positive constant used in the controller tuning. From (3.17) it is deduced that q_x converges to 0 and by consequence

$$p_x \to -K_{1x}K_{px}e_x \quad \text{as} \quad \text{t} \to \infty \tag{3.18}$$

Using this result in the first equation of (3.4), it yields

$$e_x \to 0 \tag{3.19}$$

and from (3.18)

$$p_x \to 0 \tag{3.20}$$

Therefore, the complete desired input that makes converge e_x to 0, will have the expression

$$u_x = m\ddot{r}_{dx} - mK_{1x}\sigma_{M_{1x}}\left(K_{px}p_x + K_{1x}^{-1}K_{2x}\sigma_{M_{2x}}\left(K_{qx}q_x\right)\right) \tag{3.21}$$

In a similar way, the desired input u_y which will make converge e_y to 0 has the corresponding expression

$$u_y = m\ddot{r}_{dy} - mK_{1y}\sigma_{M_{1y}}(K_{py}p_y + K_{1y}^{-1}K_{2y}\sigma_{M_{2y}}(K_{qy}q_y)) \tag{3.22}$$

where K_{1y}, M_{1y}, K_{py}, K_{1y}, K_{2y}, M_{2y}, K_{qy} are constant and used for tuning the controller. The variable q_y is defined in a similar way

$$q_y = K_{1y}K_{py}e_y + p_y \tag{3.23}$$

Once the required force to move the cart to the desired position has been found, it will be necessary to compute the thrust, pitch and yaw angles that could generate this force. First, consider $\nu = f\cos\theta$ as the force in the xy plane that makes move the vehicle. The next step is to find the desired couple (ν_d, ψ_d) that generates the desired force $\boldsymbol{u_d}$ computed in (3.21) and (3.22). Thus, from (3.3), the following equality is deduced:

$$u_x \sin\psi_d - u_y \cos\psi_d = 0 \implies \sqrt{{u_x}^2 + {u_y}^2}\sin(\psi_d + \zeta) = 0$$

The yaw angle that satisfy this equality is

$$\psi_d = -\zeta$$

The vehicle must attain this orientation. There will be two possible alternatives, that is, the vehicle orientation could be ψ_d or $\psi_d + 180°$

$$\psi_{d1} = \arg(u_x + i \cdot u_y) \tag{3.24}$$
$$\psi_{d2} = \arg(-u_x - i \cdot u_y) \tag{3.25}$$

These two equations represent the same direction of vector $\boldsymbol{u_d}$, and therefore the drone must to take the closest one to its yaw angle ψ. The direction of the thrust can be changed by means of the pitch angle θ to ensure the movement in the correct sense. This situation is described in Fig. 3.3.

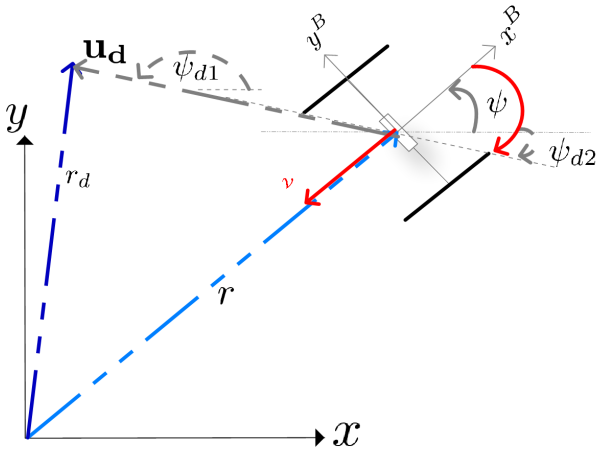


Figure 3.3: Desired yaw angle for minimal effort.

The goal is to reach the desired direction with the minimal effort. The closest angle satisfies $\cos(\psi - \psi_d) \geq 0$. Once the closest desired yaw angle is known, the expression for this angle will be rewritten in such a way that the cart will do the minimal effort to reach the desired direction. From (3.3), the following equality holds:

$$\mathbb{R}_{xy}^T \boldsymbol{u_d} = \begin{pmatrix} \nu_d \\ 0 \end{pmatrix}$$

with

$$\mathbb{R}_{xy} = \begin{pmatrix} \cos\psi_d & -\sin\psi_d \\ \sin\psi_d & \cos\psi_d \end{pmatrix}$$

Consequently,

$$u_x \cos\psi_d + u_y \sin\psi_d = \nu_d = f \cdot \sin\theta_d \tag{3.26}$$

and thence,

$$\sin\theta_d = \frac{u_x \cos\psi_d + u_y \sin\psi_d}{f} \tag{3.27}$$

which can be simplified as:

$$\sin\theta_d = \pm\frac{\sqrt{{u_x}^2 + {u_y}^2}}{f} \tag{3.28}$$

The sign depends on which ψ_d is taken. It will be positive if (3.24) is chosen or negative if (3.25) is taken. The next step is to determine the thrust f. The thrust may be chosen arbitrarily but it must satisfy the conditions in the inequality (3.29), deduced from (3.3) and (3.27), in order to be capable of generate the required force $\boldsymbol{u_d}$ and the pitch angle θ_d.

$$f \geq \sqrt{{u_x}^2 + {u_y}^2} \tag{3.29}$$

Take,

$$f = \frac{K_{min} + \rho\sqrt{{u_x}^2 + {u_y}^2}}{K_f + \cos(\psi - \psi_d)} \tag{3.30}$$

where K_{min}, K_f and ρ are positive constant. If ρ is chosen as:

$$\rho \geq K_f + 1 \tag{3.31}$$

thus, (3.30) will satisfy (3.29). The factor $K_f + \cos(\psi - \psi_d)$ will allow the cart to prioritize finding the desired ψ_d and afterwards, the cart will move to the desired trajectory with the pitch angle determined by (3.27) but it is necessary to take $K_f < 1$. Supposing $u_x = 0$, $u_y = 0$, $\psi = \psi_d$, the thrust

$$\frac{K_{min}}{K_f + \cos(\psi - \psi_d)} \tag{3.32}$$

will let the cart to compensate the dynamics not taken into account in the mathematical model when it will be at the desired position. For instance, when the gravity center of cart is not perfectly placed at the rotation center, a rotation would be produced and this trust will counteract it. The expression (3.30) must satisfy the condition in (2.1). In order to do this, it is sufficient that,

$$\frac{K_{min} + \rho\sqrt{{(u_x)_{max}}^2 + {(u_y)_{max}}^2}}{K_f} \leq mg \tag{3.33}$$

with $0 < K_f < 1$. It follows,

$$\sqrt{{(u_x)_{max}}^2 + {(u_y)_{max}}^2} \leq \frac{mg \cdot K_f - K_{min}}{\rho} \tag{3.34}$$

and also,

$$|u| \leq m|\ddot{r}_d|_{max} + m\sqrt{(K_{1x}M_{1x})^2 + (K_{1y}M_{1y})^2} \tag{3.35}$$

and making

$$m|\ddot{r}_d|_{max} + m\sqrt{(K_{1x}M_{1x})^2 + (K_{1y}M_{1y})^2} \leq \frac{mg \cdot K_f - K_{min}}{\rho} \tag{3.36}$$

then,

$$\sqrt{(K_{1x}M_{1x})^2 + (K_{1y}M_{1y})^2} \leq \frac{g \cdot K_f - (K_{min}/m)}{\rho} - |\ddot{r}_d|_{max} \tag{3.37}$$

In order to simplify the previous expression, the following assumption is taken

$$K_{1x}M_{1x} = K_{1y}M_{1y} = K_1M_1 \tag{3.38}$$

hence,

$$K_1 M_1 \leq \frac{g \cdot K_f - (K_{min}/m)}{\sqrt{2}\,\rho} - \frac{|\ddot{r}_d|_{max}}{\sqrt{2}} \tag{3.39}$$

In addition, regarding the maximum desired pitch angle, from (3.28) and (3.30), it results that:

$$|\theta_d|_{max} \leq \sin^{-1}\left(\frac{1+K_f}{\rho}\right) \tag{3.40}$$

3.1.2 Quadrotor Operation Mode

In this operation mode, the position mathematical model (2.16) becomes

$$\begin{aligned} \dot{\boldsymbol{r}} &= \boldsymbol{v} \\ m\dot{\boldsymbol{v}} &= \boldsymbol{u} \end{aligned} \tag{3.41}$$

with $\boldsymbol{v}$ as the velocity and $\boldsymbol{u}$ as the force acting on the drone. In addition,

$$\boldsymbol{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix},\ \boldsymbol{u} = \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} \tag{3.42}$$

and defining the variables $\boldsymbol{e_r}$ as the position error and $\boldsymbol{p}$ as the velocity error

$$\begin{aligned} \boldsymbol{e_r} &= \boldsymbol{r} - \boldsymbol{r_d} \\ \boldsymbol{p} &= \boldsymbol{v} - \dot{\boldsymbol{r}}_{\boldsymbol{d}} \end{aligned}$$

then, in a similar way to the methodology employed for the cart, the desired force for the quadrotor results

$$\begin{aligned} u_x &= m\ddot{r}_{dx} - mK_{1x}\sigma_{M_{1x}}\left(K_{px}p_x + K_{1x}^{-1}K_{2x}\sigma_{M_{2x}}\left(K_{qx}q_x\right)\right) \\ u_y &= m\ddot{r}_{dy} - mK_{1y}\sigma_{M_{1y}}\left(K_{py}p_y + K_{1y}^{-1}K_{2y}\sigma_{M_{2y}}\left(K_{qy}q_y\right)\right) \\ u_z &= m\ddot{r}_{dz} - mK_{1z}\sigma_{M_{1z}}\left(K_{pz}p_z + K_{1z}^{-1}K_{2z}\sigma_{M_{2z}}\left(K_{qz}q_z\right)\right) \end{aligned} \tag{3.43}$$

The following step is to find the relationship between this force, the Euler angles and thrust that can produce it. From (3.42) it can be deduced

$$\begin{aligned} f &= \sqrt{u_x^2 + u_y^2 + (u_z + mg)^2} \\ \phi_d &= \arcsin\left(\frac{u_x \sin\psi - u_y \cos\psi}{f}\right) \\ \theta_d &= \arctan\left(\frac{u_x \cos\psi + u_y \sin\psi}{u_z + mg}\right) \\ \psi_d &= \text{cte.} \end{aligned} \tag{3.44}$$

where ψ_d is the desired yaw angle and can be chosen arbitrarily. ϕ_d and θ_d are the desired roll and pitch angle respectively. The attitude control algorithm must make converge the quadrotor orientation to the desired one given by (3.44).

3.2 Nonlinear Control with Proportional Derivative Properties

This algorithm is based on a proportional derivative controller. The gains are not constant and will dependent on the distance to the position reference and on the speed the vehicle approaches this reference. This idea arises from a natural point of view. The idea is to move the vehicle in the line connecting the vehicle center to the target position, then to follow this line in order to reach its target position. The vehicle needs to get inclined in order to moving and this inclination will be proportional to the distance to the reference. This position algorithm will yield the desired attitude and thrust the vehicle must follow.

3.2.1 Cart Operation Mode

In this mode, the idea of this algorithm is to orientate the vehicle in the direction of the target position. Remember that the orientation is given by the yaw angle. After the vehicle gets the right direction, it will move to the target. This movement will be produced by the inclination of the vehicle about its y^B axis. This inclination will be given by the pitch angle and will depend on the distance from the vehicle mass center to the target position and on the speed of the movement. Bigger is the distance to the target, bigger is the inclination and consequently it is convenient to make a well-tune of the control parameters in order to limit its speed. Fig. 3.4 describes this algorithm.

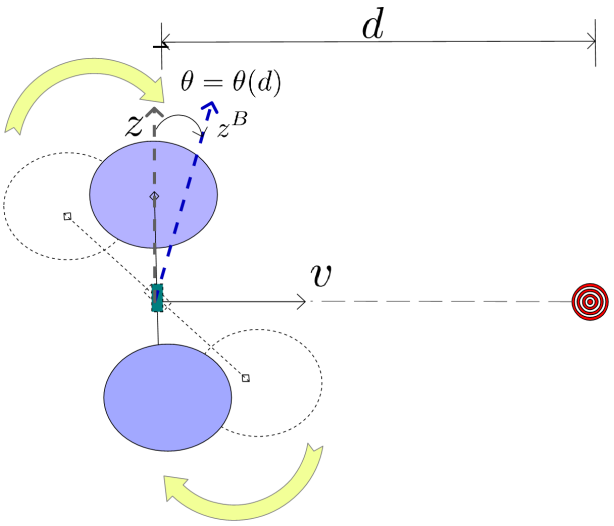


Figure 3.4: Scheme of the algorithm for the cart. First the vehicle turns around itself in order to get aligned to the target. Then, it gets inclined in order to reach the target.

The position equation from the set (2.25) considers a force term $\boldsymbol{\delta_u}$ which is not zero when the vehicle operates over the ground. This term will compensate the difference between the thrust and the vehicle weight. Under the suppositions that the vehicle moves over a perfect flat and it is always in touch with the floor, the roll angle will be zero and the height z will

be constant thus the position dynamics reduces to

$$m \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix} \cdot f \sin\theta + \begin{pmatrix} \delta_{ux} \\ \delta_{uy} \end{pmatrix} \tag{3.45}$$

where x and y stand for the xy position in an inertial frame, δ_{ux} and δ_{uy} are the xy components of $\boldsymbol{\delta_u}$. This term will represents the forces which prevent a lateral displacement in terrestrial mode. Now, let us define the error position as:

$$\boldsymbol{e_{xy}} = \begin{pmatrix} e_x \\ e_y \end{pmatrix} = \begin{pmatrix} x_{ref} - x \\ y_{ref} - y \end{pmatrix} \tag{3.46}$$

and

$$d = \|\boldsymbol{e_{xy}}\| \tag{3.47}$$

Then, defining α as the direction that the vehicle must follow and β as the inclination which produces the movement to the target, their respective expressions are proposed

$$\begin{array}{ll} \alpha & = \arg\,(e_x + \hat{\imath} \cdot e_y) \\ \beta & = \arcsin\,(k_{pg} d + k_{dg} \dot{d}) \end{array} \tag{3.48}$$

where k_{pg} and k_{dg} must be chosen in order to satisfy:

$$|\, k_{pg} d + k_{dg} \dot{d} \,| < \pi/2$$

and if the reference for the Euler angles and thrust are chosen according to (3.49), then the target position in ground mode can be achieved.

$$\begin{array}{rcl} \psi_{ref} & = & \{\alpha, \alpha + \pi : \cos(\psi_{ref} - \psi) > 0\} \\ \theta_{ref} & = & \text{sign}(\cos(\psi_{ref} - \alpha))\ \beta \\ \phi_{ref} & = & 0 \\ f & = & f_{cte} < m \cdot g \end{array} \tag{3.49}$$

f is chosen in such a way to avoid the take-off, ψ_{ref} will generate the minimal effort to align the vehicle to the target, θ_{ref} will produce an inclination always to the desired position and $\phi_{ref} = 0$ is because of the flat surface.

Stability and Convergence

In order to develop this section, first it is necessary to transform Eqn. (3.45) from Cartesian to polar coordinates. Fig. (3.5) might help to have an overall panorama. This transformation yields:

$$\begin{array}{ll} \ddot{d} - \dot{\vartheta}^2 d & = \frac{1}{m}(-f_{cte} \sin\beta \cos\gamma_g - \delta_{uxy} \sin\gamma_g) \\ d\ddot{\vartheta} + 2\dot{d}\dot{\vartheta} & = \frac{1}{m}(f_{cte} \sin\beta \sin\gamma_g - \text{sign}(\dot{\vartheta})\delta_{uxy} \cos\gamma_g) \end{array} \tag{3.50}$$

where d and ϑ represent the polar coordinates, $\delta_{uxy} = \|(\delta_{ux}\ \delta_{uy})\|$ and $\cos\gamma_g > 0$ because of (3.49).

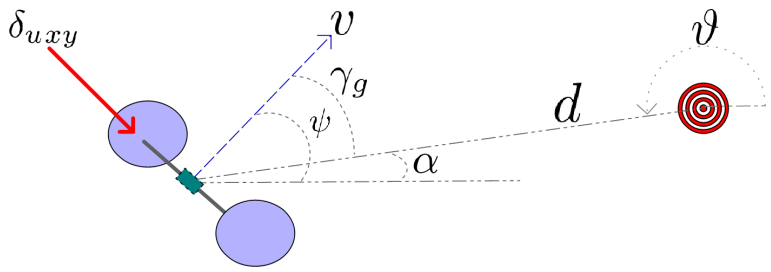


Figure 3.5: Diagram of polar coordinates of the cart. The origin is placed on the target position. There is a force δ_{uxy} which prevents a lateral displacement. γ_g represents the difference between the desired orientation α and the orientation ψ of the vehicle.

The choice of θ_{ref} in such a way the vehicle moves toward the target leads to a negative $\dot{d}$. The second equation of (3.50) can be rewritten as:

$$d\ddot{\vartheta} = \frac{1}{m}(f_{cte}\sin\beta\sin\gamma_g) - \frac{1}{m}\text{sign}(\dot{\vartheta})(\delta_{uxy}\cos\gamma_g + 2\dot{d}|\dot{\vartheta}|) \tag{3.51}$$

and if δ_{uxy} is big enough so that $\delta_{uxy}\cos\gamma_g + 2\dot{d}|\dot{\vartheta}| > 0$, this implies the angular velocity $\dot{\vartheta}$ will have the form:

$$\dot{\vartheta} = k_\vartheta \sin\beta\sin\gamma_g + o(\epsilon) \tag{3.52}$$

Additionally, γ_g will be zero when the yaw control makes $\psi = \alpha$ which implies $\dot{\vartheta} \to 0$, and it happens similarly when $\beta = 0$. This result indicates the vehicle moves straightforward to the target. As this result is based on the value of δ_{uxy}, it is important to choose a convenient material for the wheels in order to have a suitable friction with the floor.

Now, considering the expression proposed for β, the first equation of (3.50) can be rewritten as

$$\ddot{d} + a_1\ \dot{d} + a_0\ d = -\frac{1}{m}\delta_{uxy}\sin\gamma_g \tag{3.53}$$

with

$$\begin{aligned} a_0 &= \tfrac{1}{m} f_{cte}\cos\gamma_g k_{pg} - \dot{\vartheta}^2 \\ a_1 &= \tfrac{1}{m} f_{cte}\cos\gamma_g k_{dg} \end{aligned}$$

From the previous analysis we know $\dot{\vartheta} \to 0$ and $\gamma_g \to 0$. Also we know the dynamics represented by (3.53) is stable iff a_0 and a_1 are positive. Then we can conclude that $d \to 0$.

Note that if the target position is fixed, when the vehicle enters in a very small neighborhood centered at the target it will begin to turn around itself. This phenomenon is caused by the definition of α and a solution could be to make constant the yaw angle when the UGV enters into this neighborhood.

3.2.2 Quadrotor Operation Mode

In this mode, the idea is to move the vehicle in a direction given by α and with an inclination β. This part is an extension to the algorithm developed for the ground mode. In this case, there is no friction on the wheels and it is required to add a new term in order to compensate a lateral movement to the line connecting the mass center with the target position. This term will be proportional to the velocity of the vehicle. The inclination β will depend on the magnitude of the position error and on the approaching velocity to the desired position. The desired Euler angles can be generated from these α and β, see Fig. 3.6 for a graphical representation.

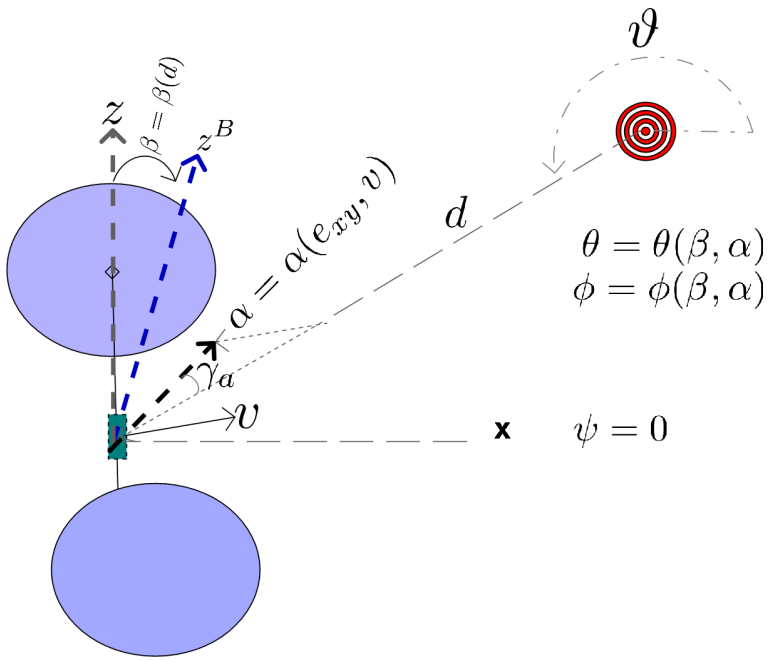


Figure 3.6: Diagram of polar coordinates for the quadrotor. The direction α tries to compensate the lateral movement. β represents the desired inclination in order to move to the target. γ_a is the difference between the direction α and the direction of the line connecting the vehicle and the target.

In aerial mode the vehicle becomes an UAV and its model reduces to (3.54)

$$m\ddot{\boldsymbol{r}} = \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} \tag{3.54}$$

The xy position error is defined in (3.46) and additionally the z position error is denoted as

$$e_z = z_{ref} - z; \tag{3.55}$$

where z represents the position of the vehicle into an inertial frame and z_{ref} is the desired

height. Then, the expressions for α, β and for the desired force u_z are proposed as

$$\begin{aligned} \alpha &= \arg\ (\ (e_x + k_v \dot{e}_x) + \hat{\imath} \cdot (e_y + k_v \dot{e}_y)\ ) \\ \beta &= \text{asin}\ (k_{pa} d + k_{da} \dot{d}) \\ u_z &= k_{pz} e_z + k_{dz} \dot{e}_z + k_{iz} \int_0^t e_z d\tau \end{aligned} \tag{3.56}$$

d is described in (3.47), k_v is a positive constant which counteracts the lateral movement, k_{pz}, k_{dz} and k_{iz} represent positive constant values used for tuning the control parameters. Similarly, k_{pa} and k_{da} are positive constant values satisfying

$$|\, k_{pa} d + k_{da} \dot{d}\,| < \pi/2$$

and by choosing the expressions in (3.57) as the references for the Euler angles and the thrust, then the target position can be achieve in aerial mode.

$$\begin{aligned} \theta_{ref} &= \text{atan}(\cos(\psi - \alpha) \tan\beta) \\ \phi_{ref} &= \text{asin}(\sin(\psi - \alpha) \sin\beta) \\ \psi_{ref} &= \psi_{cte} \\ f_a &= \frac{m \cdot g + u_z}{\cos\phi \cos\theta} \end{aligned} \tag{3.57}$$

The equations corresponding to the Euler angles can be deduced from the coordinates of the z axis when it is inclined β rads in the α direction. The value $\psi_{a,ref}$ can be assigned arbitrarily.

Stability Analysis

Firstly, the Eqn. (3.54) is transformed from Cartesian to cylindrical coordinates. Therefore, the following system results

$$\begin{aligned} \ddot{d} - \dot{\vartheta}^2 d &= \tfrac{1}{m}(-f_a \sin\beta \cos\gamma_a) \\ d\ddot{\vartheta} + 2\dot{d}\dot{\vartheta} &= \tfrac{1}{m}(f_a \sin\beta \sin\gamma_a) \\ m\ddot{z} &= f_a \cos\phi \cos\theta - m \cdot g \end{aligned} \tag{3.58}$$

Taking the expression for f_a and substituting it into the third equation of (3.58), it yields

$$m\ddot{e}_z + k_{dz}\dot{e}_z + k_{pz} e_z + k_{iz} \int_0^t e_z d\tau = 0 \tag{3.59}$$

which is stable iff $k_{pz} > m \cdot k_{iz}/k_{dz}$ and hence $e_z \to 0$. Now, the first and second equation of (3.58) can be rewritten as (3.60) and (3.61) after substituting the expression for β and taking into account that γ_a can be expressed as $\gamma_a = -|\gamma_a|\text{sign}(\dot{\vartheta})$. Hence,

$$\ddot{d} + \left(\frac{f_a}{m} \cos\gamma_a k_{da}\right)\, \dot{d} + \left(\frac{f_a}{m} \cos\gamma_a k_{pa} - \dot{\vartheta}^2\right)\, d = 0 \tag{3.60}$$

$$d\ddot{\vartheta} = -\text{sign}(\dot{\vartheta}) \left[\left(\frac{f_a}{m} \sin|\gamma_a| k_{da} + 2|\dot{\vartheta}|\right)\dot{d} + \frac{f_a}{m} \sin|\gamma_a| k_{pa} d \right] \tag{3.61}$$

Therefore, the two conditions for assuring the convergence of $d \to 0$ and $\dot{\vartheta} \to 0$ are:

$$\frac{f_a}{m} \cos \gamma_a k_{pa} - \dot{\vartheta}^2 > 0 \tag{3.62}$$

$$\frac{f_a \sin|\gamma_a| k_{pa}}{f_a \sin|\gamma_a| k_{da} + 2m|\dot{\vartheta}|} \, d > -\dot{d} \tag{3.63}$$

when $|\gamma_a| > 0$. If $|\gamma_a| = 0$ this means, $\dot{\vartheta} = 0$ and by consequence $d \to 0$. In the case, when $|\gamma_a| > 0$ it is necessary the magnitude of the approaching speed to the target does not exceed the value given by (3.63). If (3.63) holds, thus $\dot{\vartheta} \to 0$. Therefore, we need to tune the parameters k_{da} and k_{pa} in order to have an approaching speed satisfying (3.63) and consequently this leads to $d \to 0$.

Considering the case when the initial $\dot{d}_0$ be more negative than the value indicated in (3.63), this implies that $\dot{\vartheta}$ increases and thus there will be a time T when (3.62) will not be satisfied and this also leads to $\dot{d} > 0$. In other words, the tendency of $\dot{d}$ is to satisfy (3.63) and this ensures $\dot{\vartheta} \to 0$ and $d \to 0$.

3.3 Backstepping

The backstepping technique applied to control the position of a quadrotor is described in Fig. 3.7. This technique consists in computing a virtual velocity that leads the drone to the desired position. After, a virtual force is designed in order to make converge the velocity of the vehicle to the virtual one previously computed. The next step is to compute the corresponding desired Euler angles and thrust needed to generate the virtual force. After, a virtual angular velocity is designed in order to reach orientation given by the desired Euler angles. Finally, the torques that the vehicle needs to reach the virtual angular velocity.

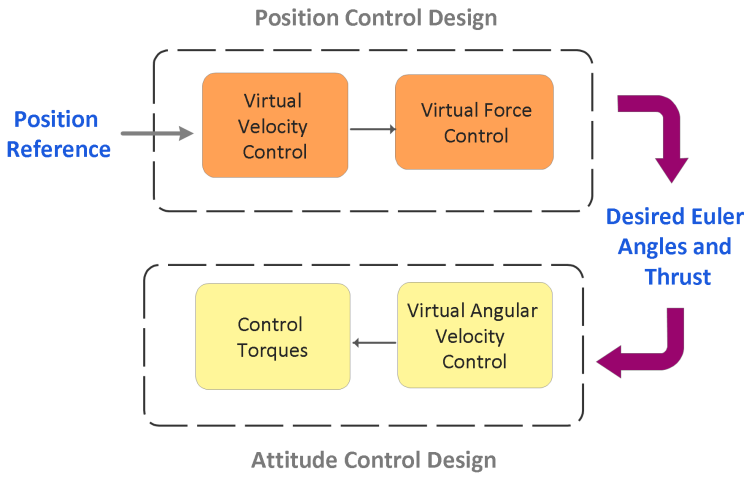


Figure 3.7: Scheme of the backstepping technique applied in the control of an UAV.

It is possible to have several adaptations of this technique. In the following, a backstepping control algorithm with integral properties and another backstepping control algorithm with

adaptive properties are developed. Somes other techniques can be found in literature [41–50] like fuzzy, model free control, adaptive PID and adaptive ones. The main goal of these algorithms is to compensate the not modeled dynamics and to counteract external perturbations. This gives the vehicle the ability to adapt itself to unknown environments.

3.3.1 Integral Properties

The robustness of the backstepping technique is wide known and, in addition, an integral part is added to ensure the convergence to desired position. The integral part helps also to reduce the position error and adds a factor that improves the robustness when the parameters of the system are not well-known.

Consider the model as in (3.41) and (3.42) and take the position error as

$$\boldsymbol{e_r} = \boldsymbol{r} - \boldsymbol{r_{ref}} \quad \Longrightarrow \quad \dot{\boldsymbol{e}}_{\boldsymbol{r}} = \dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}_{\boldsymbol{ref}} = \boldsymbol{v} - \dot{\boldsymbol{r}}_{\boldsymbol{ref}} \tag{3.64}$$

Then, a positive definite function is proposed in order to design a convenient velocity $\boldsymbol{v^v}$ that ensures the convergence to the desired position.

$$V_{Lr} = \frac{1}{2}\boldsymbol{\chi_1}^T \mathbb{K}_{Ir} \boldsymbol{\chi_1} + \frac{1}{2}\boldsymbol{e_r}^T \boldsymbol{e_r} \tag{3.65}$$

with $\mathbb{K}_{Ir}$ is a positive diagonal constant matrix that will be used for tuning the control and

$$\boldsymbol{\chi_1} = \int_0^t \boldsymbol{e_r}\ d\tau \tag{3.66}$$

Therefore

$$\dot{V}_{Lr} = \boldsymbol{\chi_1}^T \mathbb{K}_{Ir} \boldsymbol{e_r} + \boldsymbol{e_r}^T \dot{\boldsymbol{e}}_{\boldsymbol{r}} \tag{3.67}$$

Now, taking the virtual velocity control as:

$$\boldsymbol{v^v} = \dot{\boldsymbol{r}}_{\boldsymbol{ref}} - \mathbb{K}_{Ir} \boldsymbol{\chi_1} - \mathbb{K}_r \boldsymbol{e_r} \tag{3.68}$$

and thus,

$$\dot{V}_{Lr}|_{\boldsymbol{v}=\boldsymbol{v^v}} = -\boldsymbol{e_r}^T \mathbb{K}_r \boldsymbol{e_r} \quad \leq 0 \quad \forall\ t \geq 0 \tag{3.69}$$

Then, the velocity error is defined as

$$\boldsymbol{e_v} = \boldsymbol{v} - \boldsymbol{v^v} \quad \Longrightarrow \quad \dot{\boldsymbol{e}}_{\boldsymbol{v}} = \ddot{\boldsymbol{r}} - \dot{\boldsymbol{v^v}} = \frac{1}{m}\boldsymbol{u} - \dot{\boldsymbol{v^v}} \tag{3.70}$$

and let be the following positive definite function

$$V_{Lv} = V_{Lr} + \frac{1}{2}\boldsymbol{e_v}^T \boldsymbol{e_v} \tag{3.71}$$

thus

$$\dot{V}_{Lv} = \dot{V}_{Lr} + \boldsymbol{e_v}^T \dot{\boldsymbol{e}}_{\boldsymbol{v}} \tag{3.72}$$

and taking into account that $\boldsymbol{v} = \boldsymbol{v^v} + \boldsymbol{e_v}$, the expression for $\dot{V}_{Lv}$ can be written as

$$\dot{V}_{Lv} = -\boldsymbol{e_r}^T \mathbb{K}_r \boldsymbol{e_r} + \boldsymbol{e_r}^T \boldsymbol{e_v} + \boldsymbol{e_v}^T \dot{\boldsymbol{e}}_{\boldsymbol{v}} \tag{3.73}$$

and choosing the virtual control,

$$\boldsymbol{u^v} = m\left(\dot{\boldsymbol{v}^v} - \boldsymbol{e_r} - \mathbb{K}_v \boldsymbol{e_v}\right) \tag{3.74}$$

It results that

$$\dot{V}_{Lv}|_{\boldsymbol{u}=\boldsymbol{u^v}} \quad = \quad -\boldsymbol{e_r}^T \mathbb{K}_r \boldsymbol{e_r} - \boldsymbol{e_v}^T \mathbb{K}_v \boldsymbol{e_v} \leq 0 \quad \forall\, t \geq 0 \tag{3.75}$$

where $\mathbb{K}_v$ is a positive diagonal constant matrix that will be used for tuning the control. Keep in mind that $\boldsymbol{u} = \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g}$ and from (3.44), wen can compute the Euler angles $\boldsymbol{\eta_{ref}}$ and thrust f the vehicle needs to generate $\boldsymbol{u^v}$. The next phase is to design the attitude control law given these references.

Define the Euler angles error as

$$\boldsymbol{e_\eta} = \boldsymbol{\eta} - \boldsymbol{\eta_{ref}} \quad \Longrightarrow \quad \begin{array}{rcl} \dot{\boldsymbol{e}}_{\boldsymbol{\eta}} & = & \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_{\boldsymbol{ref}} \\ & = & \mathbb{B}(\boldsymbol{\eta})\boldsymbol{\omega} - \dot{\boldsymbol{\eta}}_{\boldsymbol{ref}} \end{array} \tag{3.76}$$

Eqn. (2.50) describes $\mathbb{B}(\boldsymbol{\eta})$. This matrix is not singular if and only if $\cos(\theta) \neq 0$.

Then, the next positive definite function is proposed

$$V_{L\eta} = \frac{1}{2}\boldsymbol{\chi_2}^T \mathbb{K}_{I\eta} \boldsymbol{\chi_2} + \frac{1}{2}\boldsymbol{e_\eta}^T \boldsymbol{e_\eta} \tag{3.77}$$

where

$$\boldsymbol{\chi_2} = \int_0^t \boldsymbol{e_\eta}\, d\tau \tag{3.78}$$

and $\mathbb{K}_{I\eta}$ is a positive diagonal constant matrix that will be used for tuning the control. Thus,

$$\dot{V}_{L\eta} = \boldsymbol{\chi_2}^T \mathbb{K}_{I\eta} \boldsymbol{e_\eta} + \boldsymbol{e_\eta}^T \dot{\boldsymbol{e}}_{\boldsymbol{\eta}} \tag{3.79}$$

and by choosing the virtual angular velocity $\boldsymbol{\omega^v}$,

$$\boldsymbol{\omega^v} = \mathbb{B}^{-1}\left(\dot{\boldsymbol{\eta}}_{\boldsymbol{ref}} - \mathbb{K}_{I\eta}\boldsymbol{\chi_2} - \mathbb{K}_\eta \boldsymbol{e_\eta}\right) \tag{3.80}$$

with $\mathbb{K}_\eta$ as a positive diagonal constant matrix, it yields

$$V_{L\eta}|_{\boldsymbol{\omega}=\boldsymbol{\omega^v}} = -\boldsymbol{e_\eta}^T \mathbb{K}_\eta \boldsymbol{e_\eta} \quad \leq 0 \quad \forall\, t \geq 0 \tag{3.81}$$

Now, the angular velocity error is defined as

$$\boldsymbol{e_\omega} = \boldsymbol{\omega} - \boldsymbol{\omega^v} \quad \Longrightarrow \quad \boldsymbol{\dot{e}_\omega} = \boldsymbol{\dot{\omega}} - \boldsymbol{\dot{\omega}^v} \tag{3.82}$$

and remembering that,

$$\boldsymbol{\omega} = \boldsymbol{\omega^v} + \boldsymbol{e_\omega} \qquad \boldsymbol{\dot{\omega}} = \mathbb{I}^{-1}\left(\boldsymbol{\tau} - [\boldsymbol{\omega}]^\times \mathbb{I}\boldsymbol{\omega}\right) \tag{3.83}$$

then, considering the following candidate Lyapunov function:

$$V_{L\omega} = V_{L\eta} + \frac{1}{2}\boldsymbol{e_\omega^T e_\omega} \tag{3.84}$$

then,

$$\dot{V}_{L\omega} = \dot{V}_{L\eta} + \boldsymbol{e_\omega^T \dot{e}_\omega} \tag{3.85}$$

therefore,

$$\dot{V}_{L\omega} = -\boldsymbol{e_\eta^T} \mathbb{K}_\eta \boldsymbol{e_\eta} + \boldsymbol{e_\eta^T} \mathbb{B} \boldsymbol{e_\omega} + \boldsymbol{e_\omega \dot{e}_\omega} \tag{3.86}$$

and by choosing

$$\boldsymbol{\tau} = [\boldsymbol{\omega}]^\times \mathbb{I}\boldsymbol{\omega} + \mathbb{I}\left(\boldsymbol{\dot{\omega}^v} - \mathbb{B}^T \boldsymbol{e_\eta} - \mathbb{K}_\omega \boldsymbol{e_\omega}\right) \tag{3.87}$$

it yields that,

$$\dot{V}_{L\omega} = -\boldsymbol{e_\eta^T} \mathbb{K}_\eta \boldsymbol{e_\eta} - \boldsymbol{e_\omega^T} \mathbb{K}_\omega \boldsymbol{e_\omega} \quad \leq 0 \quad \forall\, t \geq 0 \tag{3.88}$$

with $\mathbb{K}_\omega$ as a positive diagonal constant matrix used for control purposes.

3.3.2 Adaptive Properties

In this version, the vehicle is supposed to be under an external and unknown force and torque that prevent it to attain the desired position. Therefore, the mathematical model is considered similar to (2.25)

$$\begin{aligned} m\boldsymbol{\ddot{r}} &= \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} + \boldsymbol{\bar{k}_u} \\ \boldsymbol{\dot{\eta}} &= \mathbb{B}(\boldsymbol{\eta})\boldsymbol{\omega} \\ \mathbb{I}\boldsymbol{\dot{\omega}} &= \boldsymbol{\tau} - [\boldsymbol{\omega}]^\times \mathbb{I}\boldsymbol{\omega} + \boldsymbol{\bar{k}_\tau} \end{aligned} \tag{3.89}$$

where $\boldsymbol{\bar{k}_u}$ and $\boldsymbol{\bar{k}_\tau}$ represent the external and unknown force and torque, respectively. These parameters could also contain the dynamics not considered in the model. For instance, take in account that a real quadrotor is not perfectly symmetric and its behavior changes as time goes by. In order to develop the algorithm, take the position error as in Eqn. (3.64), then a positive definite function is proposed in order to design a convenient velocity $\boldsymbol{v_d}$ that ensures the convergence to the desired position.

$$V_{Lr} = \frac{1}{2}\boldsymbol{e_r^T e_r} \tag{3.90}$$

thus

$$\dot{V}_{Lr} = \boldsymbol{e_r^T \dot{e}_r} \tag{3.91}$$

and taking $\boldsymbol{v}$ as

$$\boldsymbol{v^v} = \dot{\boldsymbol{r}}_{\boldsymbol{ref}} - \mathbb{K}_r \boldsymbol{e_r} \tag{3.92}$$

it yields

$$\dot{V}_{Lr}|_{\boldsymbol{v}=\boldsymbol{v^v}} = -\boldsymbol{e_r}^T \mathbb{K}_r \boldsymbol{e_r} \quad \leq 0 \quad \forall\, t \geq 0 \tag{3.93}$$

with $\mathbb{K}_r$ as a positive diagonal matrix. Then, defining the velocity error

$$\boldsymbol{e_v} = \boldsymbol{v} - \boldsymbol{v^v} \quad \Longrightarrow \quad \dot{\boldsymbol{e}}_v = \dot{\boldsymbol{v}} - \dot{\boldsymbol{v^v}} = \frac{1}{m}(\boldsymbol{u} + \bar{\boldsymbol{k}}_{\boldsymbol{u}}) - \dot{\boldsymbol{v^v}} \tag{3.94}$$

where $\boldsymbol{u} = \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g}$. Let be the positive definite function

$$V_{Lv} = V_{Lr} + \frac{1}{2}\boldsymbol{e_v}^T \boldsymbol{e_v} \tag{3.95}$$

and consequently,

$$\dot{V}_{Lv} = \dot{V}_{Lr} + \boldsymbol{e_v}\dot{\boldsymbol{e}}_{\boldsymbol{v}} \tag{3.96}$$

Now, taking into account that $\boldsymbol{v} = \boldsymbol{v^v} + \boldsymbol{e_v}$, the expression $\dot{V}_{Lv}$ can be written as:

$$\dot{V}_{Lv} = -\boldsymbol{e_r}^T \mathbb{K}_r \boldsymbol{e_r} + \boldsymbol{e_r}^T \boldsymbol{e_v} + \boldsymbol{e_v}\dot{\boldsymbol{e}}_{\boldsymbol{v}} \tag{3.97}$$

and by choosing $\boldsymbol{u^v}$ as,

$$\boldsymbol{u^v} = -\hat{\boldsymbol{k}}_{\boldsymbol{u}} + m\left(\dot{\boldsymbol{v}}_{\boldsymbol{d}} - \boldsymbol{e_r} - \mathbb{K}_v \boldsymbol{e_v}\right) \tag{3.98}$$

with $\hat{\boldsymbol{k}}_{\boldsymbol{u}}$ as the estimate of $\bar{\boldsymbol{k}}_{\boldsymbol{u}}$, it results:

$$\dot{V}_{Lv} = -\boldsymbol{e_r}^T \mathbb{K}_r \boldsymbol{e_r} - \boldsymbol{e_v}^T \mathbb{K}_v \boldsymbol{e_v} + \frac{1}{m}\boldsymbol{e_v}\tilde{\boldsymbol{k}}_{\boldsymbol{u}} \tag{3.99}$$

where $\mathbb{K}_v$ represents a positive diagonal matrix. Then, defining the error between the estimate and the unknown disturbance as

$$\tilde{\boldsymbol{k}}_{\boldsymbol{u}} = \bar{\boldsymbol{k}}_{\boldsymbol{u}} - \hat{\boldsymbol{k}}_{\boldsymbol{u}} \tag{3.100}$$

if $\bar{\boldsymbol{k}}_{\boldsymbol{u}}$ is constant, thus, we have

$$\dot{\tilde{\boldsymbol{k}}}_{\boldsymbol{u}} = -\dot{\hat{\boldsymbol{k}}}_{\boldsymbol{u}} \tag{3.101}$$

this result will be used in the derivative of the following augmented positive definite function

$$V_{Lv2} = V_{Lv} + \frac{1}{2}\tilde{\boldsymbol{k}}_{\boldsymbol{u}}^T {\Gamma_1}^{-1} \tilde{\boldsymbol{k}}_{\boldsymbol{u}} \tag{3.102}$$

with Γ_1 as a positive diagonal matrix. Therefore,

$$\dot{V}_{Lv2} \;=\; -\boldsymbol{e_r}^T \mathbb{K}_r \boldsymbol{e_r} - \boldsymbol{e_v}^T \mathbb{K}_v \boldsymbol{e_v} + \tfrac{1}{m}\boldsymbol{e_v}^T \tilde{\boldsymbol{k}}_{\boldsymbol{u}} - \tilde{\boldsymbol{k}}_{\boldsymbol{u}} {\Gamma_1}^{-1} \dot{\tilde{\boldsymbol{k}}}_{\boldsymbol{u}} \tag{3.103}$$

and taking

$$\dot{\hat{\boldsymbol{k}}}_{\boldsymbol{u}} = \frac{1}{m}\Gamma_1 \boldsymbol{e_v} \tag{3.104}$$

as the desired dynamics for $\hat{\boldsymbol{k}}_{\boldsymbol{u}}$. Thus,

$$\dot{V}_{Lv2} \;=\; -\boldsymbol{e_r}^T \mathbb{K}_r \boldsymbol{e_r} - \boldsymbol{e_v} \mathbb{K}_v \boldsymbol{e_v} \quad \leq 0 \quad \forall\, t \geq 0 \tag{3.105}$$

The matrices $\mathbb{K}_r$, $\mathbb{K}_v$, Γ_1 are used for tunning the position control algorithm. The next step is to find the $\boldsymbol{\eta_{ref}}$ and f that generate the virtual force $\boldsymbol{u^v}$ obtained previously and from these references to design the attitude algorithm. First, the Euler angles error are defined as

$$\boldsymbol{e_\eta} = \boldsymbol{\eta} - \boldsymbol{\eta_{ref}} \quad \Longrightarrow \quad \begin{aligned} \dot{\boldsymbol{e}}_{\boldsymbol{\eta}} &= \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_{\boldsymbol{ref}} \\ &= \mathbb{B}(\boldsymbol{\eta})\boldsymbol{\omega} - \dot{\boldsymbol{\eta}}_{\boldsymbol{ref}} \end{aligned} \tag{3.106}$$

$\mathbb{B}(\boldsymbol{\eta})$ is presented in (2.50). Proposing the positive definite function

$$V_{L\eta} = \frac{1}{2}\boldsymbol{e}_{\boldsymbol{\eta}}^T\boldsymbol{e}_{\boldsymbol{\eta}} \tag{3.107}$$

and by choosing conveniently the angular velocity $\boldsymbol{\omega}$ as

$$\boldsymbol{\omega} = \boldsymbol{\omega^v} = \mathbb{B}^{-1}\left(\dot{\boldsymbol{\eta}}_{\boldsymbol{ref}} - \mathbb{K}_\eta \boldsymbol{e}_{\boldsymbol{\eta}}\right) \tag{3.108}$$

with $\mathbb{K}_\eta$ as a positive diagonal constant matrix, it yields

$$\dot{V}_{L\eta}|_{\boldsymbol{\omega}=\boldsymbol{\omega^v}} = -\boldsymbol{e}_{\boldsymbol{\eta}}^T\mathbb{K}_\eta\boldsymbol{e}_{\boldsymbol{\eta}} \quad \leq 0 \quad \forall\, t \geq 0 \tag{3.109}$$

Now, defining the angular velocity error as

$$\boldsymbol{e_\omega} = \boldsymbol{\omega} - \boldsymbol{\omega^v} \quad \Longrightarrow \quad \dot{\boldsymbol{e}}_{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}^{\boldsymbol{v}} \tag{3.110}$$

and keeping in mind that,

$$\boldsymbol{\omega} = \boldsymbol{\omega^v} + \boldsymbol{e_\omega} \quad , \quad \dot{\boldsymbol{\omega}} = \mathbb{I}^{-1}\left(\boldsymbol{\tau} - [\boldsymbol{\omega}]^\times\mathbb{I}\boldsymbol{\omega} + \bar{\boldsymbol{k}}_{\boldsymbol{\tau}}\right) \tag{3.111}$$

Then, considering the candidate Lyapunov function $V_{L\omega}$ as in Eqn. (3.84) and choosing

$$\boldsymbol{\tau} = -\hat{\boldsymbol{k}}_{\boldsymbol{\tau}} + [\boldsymbol{\omega}]^\times\mathbb{I}\boldsymbol{\omega} + \mathbb{I}\left(\dot{\boldsymbol{\omega}}_{\boldsymbol{v}} - \mathbb{B}^T\boldsymbol{e}_{\boldsymbol{\eta}} - \mathbb{K}_\omega\boldsymbol{e}_{\boldsymbol{\omega}}\right) \tag{3.112}$$

with $\hat{\boldsymbol{k}}_{\boldsymbol{\tau}}$ as the estimate of constant disturbance in attitude model, it yields

$$\dot{V}_{L\omega} \quad = \quad -\boldsymbol{e}_{\boldsymbol{\eta}}^T\mathbb{K}_\eta\boldsymbol{e}_{\boldsymbol{\eta}} - \boldsymbol{e}_{\boldsymbol{\omega}}^T\mathbb{K}_\omega\boldsymbol{e}_{\boldsymbol{\omega}} + \boldsymbol{e}_{\boldsymbol{\omega}}^T\mathbb{I}^{-1}\tilde{\boldsymbol{k}}_{\boldsymbol{\tau}} \tag{3.113}$$

with $\tilde{\boldsymbol{k}}_{\boldsymbol{\tau}}$ defined as

$$\tilde{\boldsymbol{k}}_{\boldsymbol{\tau}} = \bar{\boldsymbol{k}}_{\boldsymbol{\tau}} - \hat{\boldsymbol{k}}_{\boldsymbol{\tau}} \tag{3.114}$$

and supposing that $\bar{\boldsymbol{k}}_{\boldsymbol{\tau}}$ is constant, it results

$$\dot{\tilde{\boldsymbol{k}}}_{\boldsymbol{\tau}} = -\dot{\hat{\boldsymbol{k}}}_{\boldsymbol{\tau}} \tag{3.115}$$

Considering the augmented candidate Lyapunov function

$$V_{L\omega 2} = V_{L\omega} + \frac{1}{2}\tilde{\boldsymbol{k}}_{\boldsymbol{\tau}}^T{\Gamma_2}^{-1}\tilde{\boldsymbol{k}}_{\boldsymbol{\tau}} \tag{3.116}$$

with Γ_2 as a positive diagonal matrix. It results

$$\begin{aligned} \dot{V}_{L\omega 2} = \; & -\boldsymbol{e}_{\boldsymbol{\eta}}^T\mathbb{K}_\eta\boldsymbol{e}_{\boldsymbol{\eta}} - \boldsymbol{e}_{\boldsymbol{\omega}}^T\mathbb{K}_\omega\boldsymbol{e}_{\boldsymbol{\omega}} \\ & + \boldsymbol{e}_{\boldsymbol{\omega}}^T\mathbb{I}^{-1}\tilde{\boldsymbol{k}}_{\boldsymbol{\tau}} - \tilde{\boldsymbol{k}}_{\boldsymbol{\tau}}^T{\Gamma_2}^{-1}\dot{\hat{\boldsymbol{k}}}_{\boldsymbol{\tau}} \end{aligned} \tag{3.117}$$

Now, taking

$$\dot{\hat{\boldsymbol{k}}}_{\boldsymbol{\tau}} = \Gamma_2\mathbb{I}^{-1}\boldsymbol{e}_{\boldsymbol{\omega}} \tag{3.118}$$

as the desired dynamics for $\hat{\boldsymbol{k}}_{\boldsymbol{\tau}}$. Therefore,

$$\dot{V}_{L\omega 2} \quad = \quad -\boldsymbol{e}_{\boldsymbol{\eta}}^T\mathbb{K}_\eta e_\eta - \boldsymbol{e}_{\boldsymbol{\omega}}^T\mathbb{K}_\omega\boldsymbol{e}_{\boldsymbol{\omega}} \quad \leq 0 \quad \forall\, t \geq 0 \tag{3.119}$$

The matrices $\mathbb{K}_\eta$, $\mathbb{K}_\omega$ and Γ_2 are used for tuning the attitude controller.

3.4 Intermediary Quaternions

The quaternions are numbers with four components and from this point of view, a vector could be a quaternion with a zero component. Also, they have their own algebra and they can be used to express a rotation. They represent another alternative to the Euler angles or to the rotation matrix. They do not have the singularity of the Euler angles and imply less arithmetical operations than the rotation matrices. There is the fact that if a given quaternion corresponds to a specific rotation, then the negative of this one represents the same rotation. Thus two quaternions could represent a same rotation. In the designing of the attitude control algorithm, the unwinding phenomenon arises, that is, the drone takes sometimes the longer path in order to reach the desired attitude. This is the reason the use of quaternions needs to consider this effect and to implement an additional procedure to compensate it. The above-mentioned issues lead to look up for another method and in this work the intermediary quaternions are considered [51, 52]. This new class can also represent a rotation and has similar properties with the classical one. The idea of these intermediary quaternions is to use the full angle of the rotation instead of the half angle [53, 54]. This characteristic makes correspond a rotation matrix to only one intermediary quaternion. In the following, some preliminaries about this class of quaternions are introduced and afterward an attitude control algorithm is developed by using these intermediary quaternions.

3.4.1 Preliminaries

The rotation matrix $\mathbb{R}$ can be also expressed in function of its axis $\boldsymbol{\sigma}$ and angle ϑ of rotation. That is,

$$\mathbb{R} = \cos\vartheta\ I + (1-\cos\vartheta)\ \boldsymbol{\sigma}\boldsymbol{\sigma}^T + \sin\vartheta\ [\boldsymbol{\sigma}]^\times \tag{3.120}$$

where $[\boldsymbol{\sigma}]^\times$ stands for the skew symmetric matrix of $\boldsymbol{\sigma}$ and is defined as

$$[\boldsymbol{\sigma}]^\times = \begin{bmatrix} 0 & -\sigma_z & \sigma_y \\ \sigma_z & 0 & -\sigma_x \\ -\sigma_y & \sigma_x & 0 \end{bmatrix} \tag{3.121}$$

with $\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \sigma_z]^T$. From this representation, a unitary quaternion can be formed

$$\boldsymbol{q} = \cos\vartheta + \boldsymbol{\sigma}\sin\vartheta \tag{3.122}$$

This quaternion is related to the classical unitary rotation quaternion. The difference resides on one takes the full angle and the classical one takes the half of the rotation angle.

The intemediary quaternion can be written as well in function of a real part and a vectorial one, i.e., $\boldsymbol{q} = q_r + \boldsymbol{q_v}$. Now, from Eqn. (3.120) it can be deduced the following expressions for these parts:

$$q_r = \cos\vartheta = \frac{1}{2}(tr(\mathbb{R}) - 1) \tag{3.123}$$

$$[\boldsymbol{q_v}]^\times = \sin\vartheta\ [\boldsymbol{\sigma}]^\times = \frac{1}{2}(\mathbb{R} - \mathbb{R}^T) \tag{3.124}$$

The above mentioned expressions let us easily to determine the intermediary quaternion. This quaternion is also related to the angular velocity of the body. Taking into account that this quaternion takes the full angle, it yields

$$\dot{\boldsymbol{q}} = \boldsymbol{q} \circ \boldsymbol{\omega} \tag{3.125}$$

where $\circ$ means the mutiplication between quaternions. Observe that a vector can be seen as a quaternion with zero real part. Therefore from Eqn. (3.125), it follows

$$\dot{q}_r = -\boldsymbol{q}_v^T \boldsymbol{\omega} \tag{3.126}$$

$$\dot{\boldsymbol{q}}_{\boldsymbol{v}} = q_r \boldsymbol{\omega} + \boldsymbol{q}_{\boldsymbol{v}} \times \boldsymbol{\omega} \tag{3.127}$$

Now supposing, we have a desired rotation matrix $\mathbb{R}_d$, then the desired angular velocity satisfies

$$\frac{d}{dt}\mathbb{R}_d = \mathbb{R}_d[\boldsymbol{\omega}_d]^\times \implies [\boldsymbol{\omega_d}]^\times = \mathbb{R}_d^T \frac{d}{dt}\mathbb{R}_d \tag{3.128}$$

where $[\boldsymbol{\omega}_d]^\times$ means the skew symmetric matrix of the desired angular velocity. Alternatively, we can rewrite $\boldsymbol{\omega_d}$ in terms of quaternions as follows

$$\dot{\boldsymbol{q}}_d = \boldsymbol{q}_d \circ \boldsymbol{\omega}_d \implies \boldsymbol{\omega}_d = \bar{\boldsymbol{q}}_d \circ \dot{\boldsymbol{q}}_d \tag{3.129}$$

with $\bar{\boldsymbol{q}}$ as the conjugated quaternion of $\boldsymbol{q}$. This is useful because let us to determine $\boldsymbol{\omega}_d$ faster than in Eqn. (3.128). In the same way, the derivative of this desired angular velocity is given by

$$\dot{\boldsymbol{\omega}}_d = \|\boldsymbol{\omega}_d\|^2 + \bar{\boldsymbol{q}}_d \circ \ddot{\boldsymbol{q}}_d \tag{3.130}$$

with $\|\boldsymbol{\omega}_d\|$ stands for the magnitude of $\boldsymbol{\omega}_d$. Now, once we have introduced the intermediary quaternion, we can use it in the design of the attitude control.

3.4.2 Attitude Control Algorithm

The mathematical model in (2.28) is rewritten in terms of the intermediary quaternions, giving

$$\begin{aligned} m\ddot{\boldsymbol{r}} &= \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} \\ \dot{\boldsymbol{q}} &= \boldsymbol{q} \circ \boldsymbol{\omega} \\ \mathbb{J}\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau} - [\boldsymbol{\omega}]^\times \mathbb{J}\boldsymbol{\omega} \end{aligned} \tag{3.131}$$

where $\boldsymbol{q}$ stands for the intermediary quaternion and the others variables are defined as in previous sections. Supposing, the desired force $\boldsymbol{u}$ have been computed, then it is possible to find a pair $(\mathbb{R}_d, \boldsymbol{F}_d)$ that can produce the required force $\boldsymbol{u} = [u_x, u_y, u_z]^T$. Firstly, from $\boldsymbol{u} = \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g}$ and taking the desired thrust as $\boldsymbol{F_d} = [0, 0, f_d]^T$ it follows

$$f_d = \|\boldsymbol{u} - \boldsymbol{F_g}\| \tag{3.132}$$

Now, defining the desired rotation matrix as

$$\mathbb{R}_d = [\boldsymbol{R_{d1}}, \boldsymbol{R_{d2}}, \boldsymbol{R_{d3}}] \tag{3.133}$$

where $\boldsymbol{R_{di}}$ is a column vector. Then, it can be deduced that

$$\boldsymbol{R_{d3}} = \begin{bmatrix} R_{d3,x} \\ R_{d3,y} \\ R_{d3,z} \end{bmatrix} = \frac{1}{f_d}(\boldsymbol{u} - \boldsymbol{F_g}) \tag{3.134}$$

Now, by using the expression for the rotation matrix in the zyx convention for the Euler angles, the following expressions for $\boldsymbol{R_{d1}}$ and $\boldsymbol{R_{d2}}$ can be obtained

$$\boldsymbol{R_{d1}} = \begin{bmatrix} R_{d1,x} \\ R_{d1,y} \\ R_{d1,z} \end{bmatrix} = \begin{bmatrix} \cos\psi_d \cos\theta_d \\ \sin\psi_d \cos\theta_d \\ -\sin\theta_d \end{bmatrix} \tag{3.135}$$

$$\boldsymbol{R_{d2}} = \boldsymbol{R_{d3}} \times \boldsymbol{R_{d1}} \tag{3.136}$$

with

$$\cos\theta_d = \left(1 + \left[\frac{1}{R_{d3,z}}(R_{d3,x}\cos\psi_d + R_{d3,y}\sin\psi_d)\right]^2\right)^{-1} \tag{3.137}$$

$$\sin\theta_d = \cos\theta_d \left[\frac{1}{R_{d3,z}}(R_{d3,x}\cos\psi_d + R_{d3,y}\sin\psi_d)\right] \tag{3.138}$$

where ψ_d is the desired yaw angle.

Once the desired rotation matrix is computed $\mathbb{R}_d$, it is designed an orientation control algorithm which follows the desired rotation matrix. In order to compensate the effect of unknown dynamics in our simplified model, an integral part is added to the controller. Define the rotation error matrix as

$$\mathbb{R}_d = \mathbb{R}\mathbb{R}_e \quad \Longrightarrow \quad \mathbb{R}_e = \mathbb{R}^T\mathbb{R}_d \tag{3.139}$$

This error matrix represents the orientation the vehicle needs to move from its current attitude to reach the desired one. From (3.139) and using (3.121) and (3.124), the error intermediate quaternion is computed as

$$q_{e,r} = \frac{1}{2}\left(\ tr(\mathbb{R}_e) - 1\ \right) \tag{3.140}$$

$$\boldsymbol{q_{e,v}} = \frac{1}{2}\begin{bmatrix} \mathbb{R}_{e3,2} - \mathbb{R}_{e2,3} \\ \mathbb{R}_{e1,3} - \mathbb{R}_{e3,1} \\ \mathbb{R}_{e2,1} - \mathbb{R}_{e1,2} \end{bmatrix} \tag{3.141}$$

The control goal is to make converge $\mathbb{R}_e \to I$ or equivalently $\boldsymbol{q_e} \to 1 + \boldsymbol{0}$. The following positive function is suggested

$$\begin{aligned} V_q &= \tfrac{1}{2}(1 - \boldsymbol{q_e}) \circ (1 - \bar{\boldsymbol{q}}_{\boldsymbol{e}}) + \tfrac{1}{2}\ \boldsymbol{\chi_2}^T \mathbb{K}_{iq} \boldsymbol{\chi_2} \\ &= \tfrac{1}{2}(1 - q_{e,r})^2 + \tfrac{1}{2}\boldsymbol{q_{e,v}}^T \boldsymbol{q_{e,v}} + \tfrac{1}{2}\ \boldsymbol{\chi_2}^T \mathbb{K}_{iq} \boldsymbol{\chi_2} \end{aligned} \tag{3.142}$$

with $\mathbb{K}_{iq}$ as a positive constant diagonal matrix and

$$\boldsymbol{\chi_2} = \int_0^t \boldsymbol{q_{e,v}}\ d\tau$$

By differentiating (3.142), it follows

$$\dot{V}_q = (1 - q_{e,r})(-\dot{q}_{e,r}) + \boldsymbol{q}_{\boldsymbol{e,v}}^T \dot{\boldsymbol{q}}_{\boldsymbol{e,v}} + \boldsymbol{\chi}_{\mathbf{2}}^T \mathbb{K}_{iq} \boldsymbol{q}_{\boldsymbol{e,v}} \tag{3.143}$$

and by using Eqns. (3.126) and (3.127), it yields

$$\dot{V}_q = \boldsymbol{q}_{\boldsymbol{e,v}}^T \boldsymbol{\omega}_{\boldsymbol{e}} + \boldsymbol{q}_{\boldsymbol{e,v}}^T \mathbb{K}_{iq} \boldsymbol{\chi}_{\mathbf{2}} \tag{3.144}$$

where $\boldsymbol{\omega_e}$ is the error angular error rate which satisfies

$$\frac{d}{dt}\mathbb{R}_e = \mathbb{R}_e [\boldsymbol{\omega}_{\boldsymbol{e}}]^\times \tag{3.145}$$

In order to find the relationship between $\boldsymbol{\omega_e}$, $\boldsymbol{\omega}$ and $\boldsymbol{\omega_d}$, Eqn. (3.139) is differentiated and by using $\dot{\mathbb{R}} = \mathbb{R}[\boldsymbol{\omega}]^\times$ and Eqns. (3.128) and (3.145), it results

$$[\boldsymbol{\omega}_{\boldsymbol{e}}]^\times = [\boldsymbol{\omega}_{\boldsymbol{d}}]^\times - \mathbb{R}_e^T [\boldsymbol{\omega}]^\times \mathbb{R}_e \tag{3.146}$$

Alternatively, a vectorial relationship is deduced between these angular velocities. For that purpose, consider the Diagram 3.8. Observe that the angular velocities belong to two different systems.

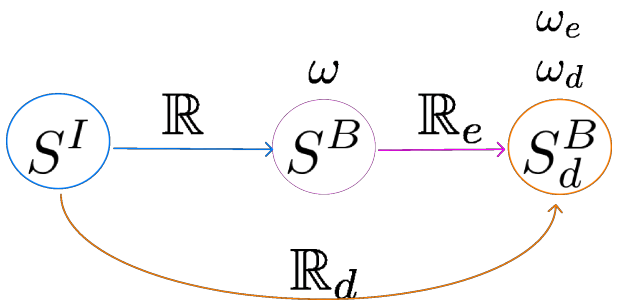


Figure 3.8: Systems generated by the rotation matrices. S^I stands for the inertial system, S^B represents the body system and S_d^B indicates the desired body orientation.

Now, taking into account that $\mathbb{R}_e[\boldsymbol{\omega}]^\times \mathbb{R}_e$ is the equivalent matrix in the system S_d^B to $[\boldsymbol{\omega}]^\times$ in the system S^B, the following vectorial relationship is deduced

$$\boldsymbol{\omega}_{\boldsymbol{e}} = \boldsymbol{w}_{\boldsymbol{d}} - \mathbb{R}_e^T \boldsymbol{\omega} \tag{3.147}$$

Then, a virtual angular rate error is proposed and which makes converge the vehicle to the desired attitude.

$$\boldsymbol{\omega}_{\boldsymbol{e}}^{\boldsymbol{v}} = -\mathbb{K}_q \boldsymbol{q}_{\boldsymbol{e,v}} - \mathbb{K}_{iq} \boldsymbol{\chi}_{\mathbf{2}} \tag{3.148}$$

where $\mathbb{K}_q$ is a constant positive diagonal matrix for tuning the algorithm. This virtual velocity leads to

$$\dot{V}_q|_{\boldsymbol{\omega}_e = \boldsymbol{\omega}_e^v} = -\boldsymbol{q}_{\boldsymbol{e,v}}^T \mathbb{K}_q \boldsymbol{q}_{\boldsymbol{e,v}} \leq 0 \tag{3.149}$$

Observe from Eqn. (3.149) that Eqn. (3.142) is a nonincreasing function and that $\boldsymbol{q_e}$ converges to two possible values, that is, $\boldsymbol{q_e} \to \boldsymbol{q_{e1}} = -1 + \mathbf{0}$ or $\boldsymbol{q_e} \to \boldsymbol{q_{e2}} = 1 + \mathbf{0}$. In order to prove that the equilibrium point $\boldsymbol{q_{e1}}$ is unstable and that $\boldsymbol{q_e} \to \boldsymbol{q_{e2}}$, we suppose firstly that $\boldsymbol{q_e} \to \boldsymbol{q_{e1}}$.

Once $\boldsymbol{q_e}$ has attained its equilibrium, it happens a small perturbation on the system which causes a small $\boldsymbol{q_{e,v}}$. This small $\boldsymbol{q_{e,v}}$ will make decrease V_q until $\boldsymbol{q_e}$ attains again its equilibrium. Repeating again the perturbation it will drive to decrease V_q. As $V_q \geq 0$, then $V_q \to 0$ which is a contradiction because the minimal value of $V_q|_{\boldsymbol{q_e}=\boldsymbol{q_{e1}}} = 2$. Then $\boldsymbol{q_{e1}}$ is an unstable equilibrium point and therefore $\boldsymbol{q_e} \to 1 + \mathbf{0}$.

Now, it is necessary to make converge $\boldsymbol{\omega_e} \to \boldsymbol{\omega_e^v}$. From (3.147), it results

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{e}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{d}} - \mathbb{R}_e^T \dot{\boldsymbol{\omega}} + \boldsymbol{\omega_e} \times \mathbb{R}_e^T \boldsymbol{\omega} \tag{3.150}$$

and introducing the dynamics of $\boldsymbol{\omega}$ from (3.131) into (3.150), it yields

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{e}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{d}} - \mathbb{R}_e^T \mathbb{J}^{-1}(\boldsymbol{\tau} - [\boldsymbol{\omega}]^{\times} \mathbb{J} \boldsymbol{\omega}) + \boldsymbol{\omega_e} \times \mathbb{R}_e^T \boldsymbol{\omega} \tag{3.151}$$

Propose an extended positive function

$$V_{q\omega} = V_q + \frac{1}{2} \boldsymbol{e}_{\boldsymbol{\omega_e}}^T \boldsymbol{e}_{\boldsymbol{\omega_e}} \tag{3.152}$$

with $\boldsymbol{e}_{\boldsymbol{\omega_e}} = \boldsymbol{\omega_e} - \boldsymbol{\omega_e^v}$. By differentiating $V_{q\omega}$, it follows

$$\dot{V}_{q\omega} = -\boldsymbol{q}_{\boldsymbol{e,v}}^T \mathbb{K}_q \boldsymbol{q_{e,v}} + \boldsymbol{q}_{\boldsymbol{e,v}}^T \boldsymbol{e}_{\boldsymbol{\omega_e}} + \boldsymbol{e}_{\boldsymbol{\omega_e}}^T \dot{\boldsymbol{e}}_{\boldsymbol{\omega_e}} \tag{3.153}$$

and by taking $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau} = [\boldsymbol{\omega}]^{\times} \mathbb{J} \boldsymbol{\omega} + \mathbb{J} \mathbb{R}_e \left(\boldsymbol{\omega_e} \times \mathbb{R}_e^T \boldsymbol{\omega} + \dot{\boldsymbol{\omega}}_{\boldsymbol{d}} - \tilde{\boldsymbol{\tau}} \right) \tag{3.154}$$

it yields

$$\dot{\boldsymbol{e}}_{\boldsymbol{\omega_e}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{e}} - \dot{\boldsymbol{\omega}}_{\boldsymbol{e}}^{\boldsymbol{v}} = \tilde{\boldsymbol{\tau}} - \dot{\boldsymbol{\omega}}_{\boldsymbol{e}}^{\boldsymbol{v}} \tag{3.155}$$

and by choosing

$$\tilde{\boldsymbol{\tau}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{e}}^{\boldsymbol{v}} - \boldsymbol{q_{e,v}} - \mathbb{K}_{\omega_e} \boldsymbol{e}_{\boldsymbol{\omega_e}} \tag{3.156}$$

then (3.153) becomes

$$\dot{V}_{q\omega} = -\boldsymbol{q}_{\boldsymbol{e,v}}^T \mathbb{K}_q \boldsymbol{q_{e,v}} - \boldsymbol{e}_{\boldsymbol{\omega_e}}^T \mathbb{K}_{\omega_e} \boldsymbol{e}_{\boldsymbol{\omega_e}} \leq 0 \tag{3.157}$$

with $\mathbb{K}_{\omega_e}$ as constant positive diagonal matrix. Eqn. (3.157) implies that $\boldsymbol{\omega_e} \to 0$ and $q_e \to 1 + \mathbf{0}$. This as well implies $\mathbb{R}_e \to I$ and $\boldsymbol{\omega} \to \mathbb{R}_e \boldsymbol{\omega_d}$ which in turn leads to $\boldsymbol{\omega} \to \boldsymbol{\omega_d}$. After some substitutions, the complete expression for $\boldsymbol{\tau}$ becomes

$$\begin{aligned} \boldsymbol{\tau} = \quad & [\boldsymbol{\omega}]^{\times} \mathbb{J} \boldsymbol{\omega} + \mathbb{J} \mathbb{R}_e \{ \boldsymbol{\omega_e} \times \mathbb{R}_e^T \boldsymbol{\omega} + \dot{\boldsymbol{\omega}}_{\boldsymbol{d}} \\ & + (q_{e,r} \mathbb{K}_q + \mathbb{K}_{\omega_e}) \boldsymbol{\omega_e} + (I + \mathbb{K}_{iq} + \mathbb{K}_{\omega_e} \mathbb{K}_q) \boldsymbol{q_{e,v}} \\ & \mathbb{K}_q (\boldsymbol{q_{e,v}} \times \boldsymbol{\omega_e}) + \mathbb{K}_{iq} \mathbb{K}_{\omega_e} \boldsymbol{\chi_2} \} \end{aligned} \tag{3.158}$$

3.5 External disturbance observer

The design and kind of observer will depend on the question we want to resolve [55–57]. In particular, the estimation of the external disturbances applied to the vehicle could be used to know its operation mode. In other words, the observer has to identify the forces acting over

the vehicle, for example, the force due to the ground reaction. This knowledge will allow to switch between the two operation modes: ground and aerial. Among the class of observers, the high order observer is preferred because it will allow to follow disturbances varying in time and the computation cost is lesser than implementing a Kalman filter for instance. The design of the observer is adapted from the work [58] which is based in a generalization of the estimation of friction forces. The observer requires a system of the following form

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t) + M\boldsymbol{d}(t) \tag{3.159}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ indicates the system state, $\boldsymbol{u} \in \mathbb{R}^m$ stands for the control input, $\boldsymbol{d} \in \mathbb{R}^r$ is a constant disturbance, $f(\cdot)$ represents the system dynamics and M is a matrix of rank r. Then, if the observer has the form (3.160), the estimate converge to the constant disturbance.

$$\begin{aligned} \hat{\boldsymbol{d}} &= \Gamma_0(M^+\boldsymbol{x} - \boldsymbol{z}) \\ \dot{\boldsymbol{z}} &= M^+ f(\boldsymbol{x}, \boldsymbol{u}, t) + \hat{\boldsymbol{d}} \end{aligned} \tag{3.160}$$

where M^+ is the pseudo-inverse of M. Now, in order to extend the application of this observer to a ramp disturbance, if the matrices $\Gamma_1 = \text{diag}(\gamma_{11}, ..., \gamma_{1r})$ and $\Gamma_0 = \text{diag}\,(\gamma_{01}, ..., \gamma_{0r})$ are chosen such that the polynomials $p_j(s) := s^2 + \gamma_{0j}s + \gamma_{1r}$, with $j \in [1, r]$, are Hurwitz stable, then the observer (3.161) converges to the ramp disturbance.

$$\hat{\boldsymbol{d}} = \Gamma_0(M^+\boldsymbol{x} - \boldsymbol{z}) + \Gamma_1 \int_0^t (M^+\boldsymbol{x} - \boldsymbol{z})d\tau \tag{3.161}$$

In the same way, a generalization can be made in order to design an observer of high order disturbances $\boldsymbol{d}(t) = \sum_{k=0}^{q} \boldsymbol{d_k} t^k$ where $\boldsymbol{d_k}$ is constant but unknown. Fig.(3.9) describes this generalization.

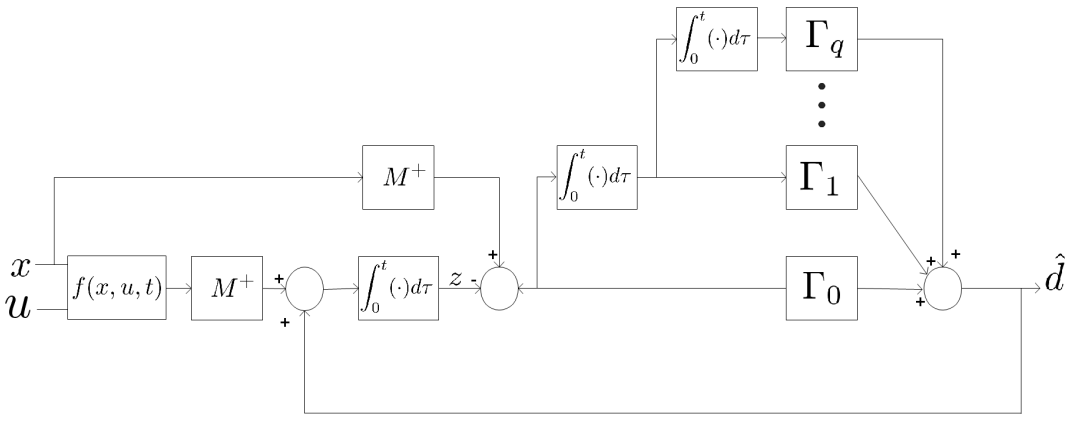


Figure 3.9: Observer scheme for a high order disturbance.

The mathematical model of the position of a quadrotor can be expressed as

$$m\ddot{\boldsymbol{r}} = \boldsymbol{u} + \boldsymbol{\delta_u} \tag{3.162}$$

where $\boldsymbol{u} = \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g}$, m stands for the vehicle mass, and $\boldsymbol{\delta_u}$ represents the external disturbance. Rewritting this model in a state space representation, it yields

$$\frac{d}{dt}\begin{pmatrix} \boldsymbol{r} \\ \dot{\boldsymbol{r}} \end{pmatrix} = \begin{pmatrix} 0_3 & I_3 \\ 0_3 & 0_3 \end{pmatrix}\begin{pmatrix} \boldsymbol{r} \\ \dot{\boldsymbol{r}} \end{pmatrix} + \begin{pmatrix} 0_3 \\ \frac{1}{m}I_3 \end{pmatrix}\boldsymbol{u} + \begin{pmatrix} 0_3 \\ I_3 \end{pmatrix}\boldsymbol{d} \tag{3.163}$$

where $\boldsymbol{d} = \boldsymbol{\delta_u}/m$. From (3.163), it follows

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{r} \\ \dot{\boldsymbol{r}} \end{pmatrix}, \qquad f(\boldsymbol{x}, \boldsymbol{u}, t) = \begin{pmatrix} 0_3 & I_3 \\ 0_3 & 0_3 \end{pmatrix} + \begin{pmatrix} 0_3 \\ \frac{1}{m} I_3 \end{pmatrix} \boldsymbol{u}$$
$$M = \begin{pmatrix} O_3 \\ I_3 \end{pmatrix}, \quad M^+ = (0_3 \quad I_3) \tag{3.164}$$

then, the number of matrices Γ_k to use will depend on the type of disturbance and on the adaptation characteristics the observer must have. A direct application of this estimator will be the switching from the aerial control algorithm to the terrestrial one in the case of the hybrid vehicle.

3.5.1 Disturbance Observer Implemented in the Control Algorithm

The results from this observer could be used in the control loop in order to improve the performance of the quadrotor. Supposing the system is already controlled and exhibits an stable behavior and assuming the system is behaving as desired, then it will be valid to represent it as in Fig.3.10.

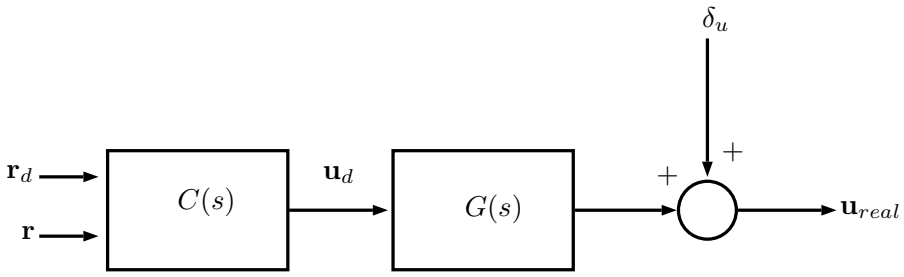


Figure 3.10: Diagram of the perturbation acting on a quadrotor subject to a position control algorithm.

From figure $\boldsymbol{r_d}$ stands for the position reference, $\boldsymbol{u_d}$ is de desired force generated by the controller $C(s)$, $G(S)$ represents the dynamics of the system when it is under control, $\boldsymbol{u_{real}}$ is the real force the vehicle undergoes and $\boldsymbol{\delta_u}$ denotes the not modeled dynamics and also an external perturbation. The system follows the desired input $\boldsymbol{u_d}$ but there is something that prevents to reach completely its goal. Thus, it is natural to think in using the estimation of the disturbance in order to counteract it.

The implementation of this estimation is described in Fig. 3.11. Supposing the dynamics $G(s)$ is known, the real force will be given by

$$\boldsymbol{u_{real}}(s) = G(S)\boldsymbol{u_d} + \left(\boldsymbol{\delta_u} - G(s)\hat{G}(s)\hat{\boldsymbol{\delta_u}}\right) \tag{3.165}$$

hence, the performance achieved will depend on the estimation $\hat{\boldsymbol{\delta_u}}$ and on the knowledge of the system $G(s)$. For instance, the estimation of the plant could be made by the help of the software MATLAB® using the pair $(\boldsymbol{u_d}, \boldsymbol{u_{real}} - \hat{\boldsymbol{\delta_u}})$.

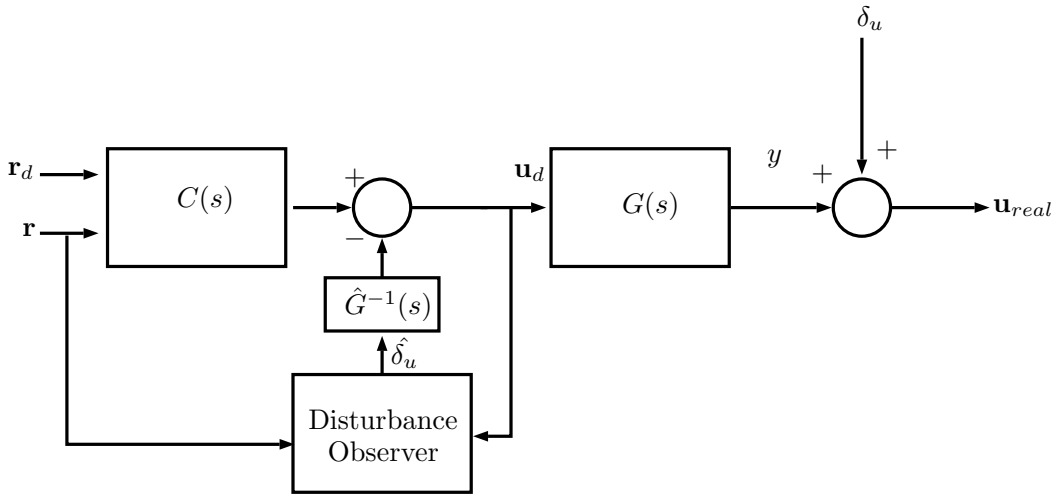


Figure 3.11: Implementation of the estimation in the control loop. The performance lies in level knowledge of $G(s)$ and of $\boldsymbol{\delta_u}$

3.6 Summary

Some techniques are developed for controlling the position of the hybrid vehicle in each mode of operation. Nested saturation algorithm let us to design slow displacement if needed and a version for each operation mode was computed. Similarly, from a natural perspective, a non linear control algorithm was proposed, in ground mode the natural friction helps to stabilize and make converge the vehicle to the target but in air mode it was necessary to implement an velocity term in the algorithm in order to provide some damping to the movement in direction to the target. These two control algorithms in particular allow to have a same kind of control with different values for their tunning parameters. The technique based on intermediary quaternion was developed for the aerial mode with the goal to avoid the singularity that appears when using Euler angles and the unwinding phenomenon when using classical quaternions. Finally, in order to identify the moment when the vehicle touches the ground, it was adapted the formulation of a high order observer to our system. Furthermore, this same observer could be used to improve the results of control algorithm. Next chapter is dedicated to validate numerically these algorithms.

Chapter 4

Numerical Validation

Contents

In this section, the control algorithms developped previously are validated by simulations using the software Simulink from MATLAB®. The saturation and the nonlinear controls are tested in terrestrial mode for generating the control force required for moving the vehicle to the desired position. In terrestrial mode, the drone is supposed to have an algorithm which follows the Euler angles and thrust required for generating the control force. In aerial operation, these same control techniques are validated and additionally, the vehicle attitude is controlled by an algorithm based on the backstepping technique. Moreover, the position control of a quadrotor using the intermediary quaternions is proved and proposed as an alternative to Euler angles and quaternion representations, this approach allows to control the vehicle attitude without the singularity issue observed with the Euler angles and without the unwinding phenomenon presented in the classical quaternions. Also, the disturbance observer developed before is used to observe the external forces that could affect the performance of position control. Finally, the saturation and nonlinear controls are tested numerically in the following of a hybrid aerial and terrestrial trajectory.

4.1 Ground Vehicle

In this mode the hybrid vehicle behaves as cart. The thrust must be regulated in order to avoid the take-off and as the vehicle is supposed to touch always the ground, the roll angle must be zero. The forces required for controlling the cart must be produced by the helices of the quadrotor. This control force will depend on the pitch angle, the thrust and the yaw angle. The yaw angle will give the orientation of the vehicle and the pitch angle will produce the forward or backward movement. The saturation technique suits because it is possible to constraint the maximum force generated by the helix rotation and by consequence it is possible to avoid the take-off. Similarly,the thrust chosen in the nonlinear technique is constant and allows to avoid the take-off as well.

4.1.1 Nested Saturations

The parameters used for the simulation of this vehicle in this mode are shown in the Table 4.1. This simulation consists in following a circular trajectory at a rate of 1/10 hertz and with a radio of $1m$. The initial conditions are zero. According to the parameters, the maximum value for $K_1M_1 \approx 1.68$. The values for the axis x and y are the same which means $K_1 = K_{1x} = K_{1y}$ and similarly for the other parameters. Additionally, in order to test the robustness of the control, the simulation considers the presence of an external perturbation of value $[0.125, 0.075]N$ and also it is supposed a white noise on the position measure.

In Fig. 4.1 the vehicle follows the trajectory but there is an error due to the external forces acting on the cart. This shows the robustness in presence of disturbances. Fig 4.2 presents the control forces required for moving the cart to the desired position. The signals are noisy because of the noise added to the position measurement. These forces are under the maximum value of K_1M_1 in order to avoid the take-off. Fig 4.3 describes the evolution of the thrust required and its maximum value is under the weight of the vehicle. The cart must have an almost constant pitch in order to follow the circular path as shown in Fig. 4.4. Finally the Fig. 4.5 describes the orientation of the cart, this angle takes values from $-180°$ to $180°$ because the vehicle is tracking a circular path.

Table 4.1: Simulation parameters for saturation control in ground mode.

Parameter	Value	Parameter	Value
mass	0.350 kg	K_f	0.5
K_1	1	M_1	1
K_2	1	M_2	0.5
K_p	1	K_q	3
ρ	2	K_{min}	0.05

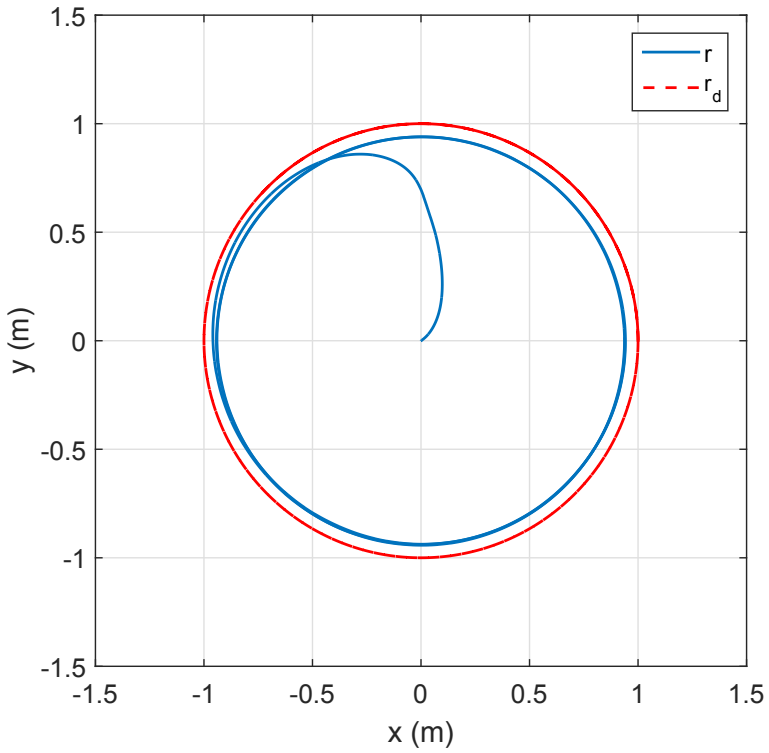


Figure 4.1: Following of a circular trajectory in ground mode using a nested saturation position control. The gap between the two trajectories is caused by the disturbances added to the model.

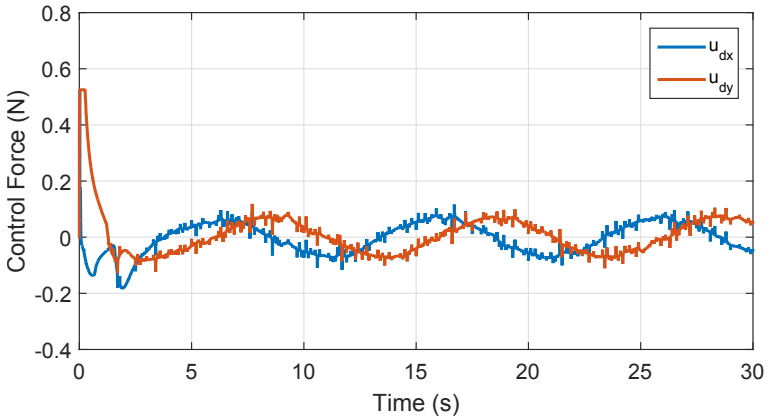


Figure 4.2: Control force generated by the nested saturation control in ground mode. The noisy appearance is due to the noise considered in the position measurement.

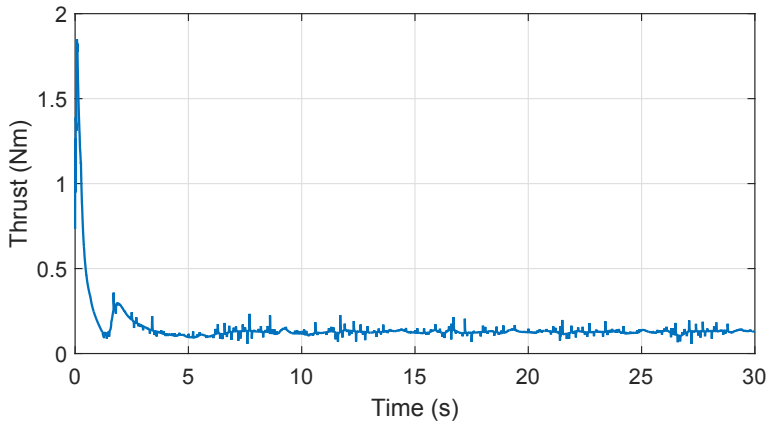


Figure 4.3: Thrust generated by the helices of the quadrotor in ground mode. This thrust do not exceed the total weight of the vehicle.

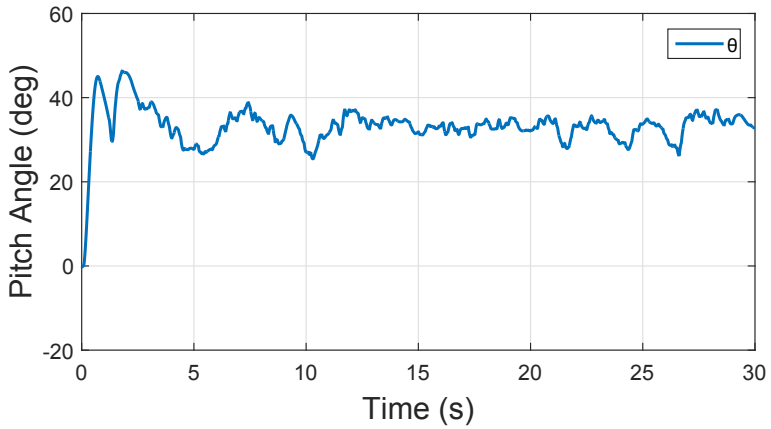


Figure 4.4: Pitch angle required by the cart in order to follow the circular path.

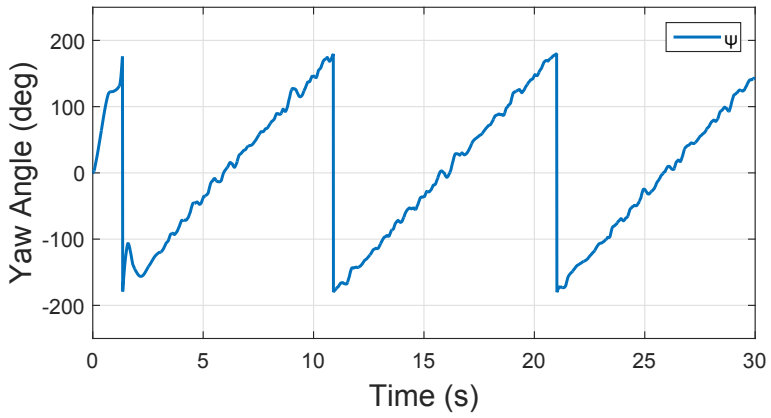


Figure 4.5: Evolution of the cart orientation during a circular path tracking.

4.1.2 Non-linear Technique

The parameters used in this algorithm are shown in Table 4.2. The mission was the same as previouly proposed for the nested saturated controller. The following of a circular path is simulated and the turning rate of the trajectory is 1/10 hertz with a radio of $1m$. The initial conditions for the position are zero. A constant thrust of $3N$ is used and the vehicle needs to take an certain pitch angle in order to track the trajectory. The orientation is given by the yaw angle which comes from the vehicle pointing to the desired position at every instant. Furthermore, in order to prove the algorithm robustness, a disturbance is added to the vehicle with value $[0.125, 0.075]N$ and also a white noise in the position measure.

Table 4.2: Simulation parameters for nonlinear control in ground mode.

Parameter	Value	Parameter	Value
mass	0.350 kg	K_p	1
K_d	6	f	3 N

In Fig. 4.6 the trajectory described by the cart is introduced. There is a gap between the two paths because of the external perturbations added in the simulation. Considering the thrust $f = 3N$, a small pitch angle is required in order to make the tracking of the circular path, as can be seen in Fig. 4.7. The vehicle orientation is described in Fig. 4.8 which agrees with the expected behavior during the following.

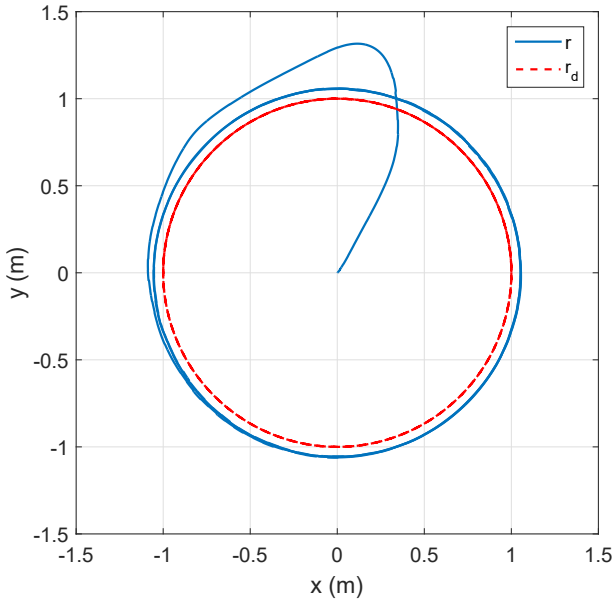


Figure 4.6: Following of a circular path by using a nonlinear control in ground mode. The thrust is constant and the algorithm must control the pitch angle in order to track the trajectory. The addition of an external force in simulation creates a gap between the two trajectories.

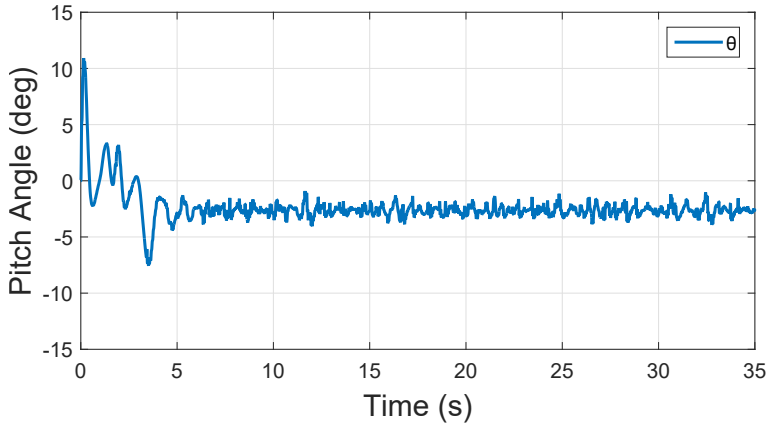


Figure 4.7: Pitch angle generated by the nonlinear algorithm. This angle is small because of the thrust value. The signal is noisy as a result of the noise added in the position measurement.

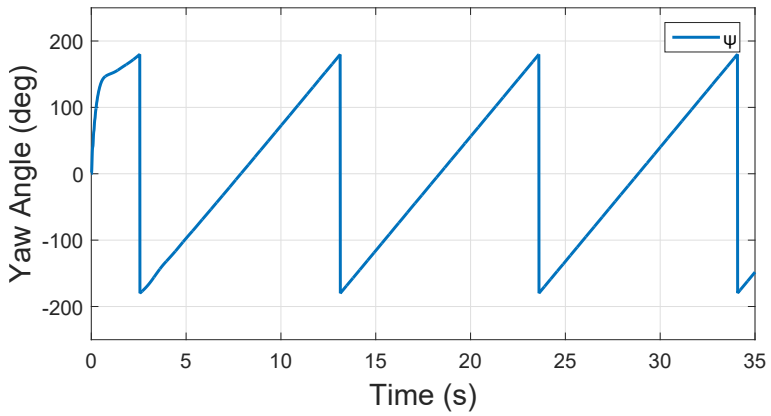


Figure 4.8: Yaw angle described by the cart when following a circular trajectory.

4.2 Quadrotor Vehicle

In aerial mode the vehicle behaves as a quadcopter and the control algorithm is adapted to its dynamics. In previous chapter, it was developed two versions of the backstepping technique. That with integral properties aims to counteract the not modeled dynamics and the disturbances that could be present during the operation. The other variant has adaptive characteristics which helps to cope the undesired effects of perturbations. The control design is divided in two parts, the first one is the position control and the second one is the attitude control. From this, it is possible to use the attitude control part based on a backstepping technique and to combine it with another position control algorithm based either on saturations or on the nonlinear approach developed beforehand. Moreover, the use of the estimation of a disturbance observer could help to improve the performance of the control law in order to reduce the negatives effects of external perturbations. This section includes a simulation of these approaches for the position control of a quadrotor.

4.2.1 Backstepping with Integral Properties

In the simulation, the quadrotor must follow a circular path at $0.6\,rad/s$ with a radio of $1\,m$ and at a height of $1\,m$. The desired yaw angle is $\psi_d = 0$ and the initial conditions are zero for the position and for the Euler angles. For the purpose of testing the robustness, an external force $\boldsymbol{\delta_u} = [0.125, 0.075, -0.125]N$ is added as a perturbation to the position model and also another one $\boldsymbol{\delta_\tau} = [0.025, 0.015, -0.025]N \cdot m$ to the orientation model. In addition, a white noise is added to the position and to the Euler angles in order to emulate real measurements. The simulation parameters are shown in Table 4.3.

Table 4.3: Simulation parameters for position control of a quadrotor based on an integral backstepping technique.

Parameter	Value	Parameter	Value
mass	0.350 kg	$\mathbb{K}_{Ir}$	diag(1, 1, 0.5)
$\mathbb{K}_r$	diag(2, 2, 2)	$\mathbb{K}_v$	diag(2, 2, 4)
$\mathbb{K}_{I\eta}$	diag(16, 16, 16)	$\mathbb{K}_\eta$	diag(4, 4, 4)
$\mathbb{K}_\omega$	diag(8, 8, 8)	$\mathbb{I}$	diag(0.008, 0.008, 0.006) kg·m^2

In Fig. 4.9, the quadrotor trajectory converges the reference path. The thrust and the Euler angles are depicted in Figs. 4.10 and 4.11. The roll and pitch angles are varying because of the inclination needed to following the circular path whereas the yaw angle is constant. Fig. 4.12 shows the control torques the vehicle need to produce the thrust and to follow the desired attitude. The integral part in the position model compensates the external forces acting as disturbances and in a same way the integral part in the attitude model counteracts the external torques added to the system.

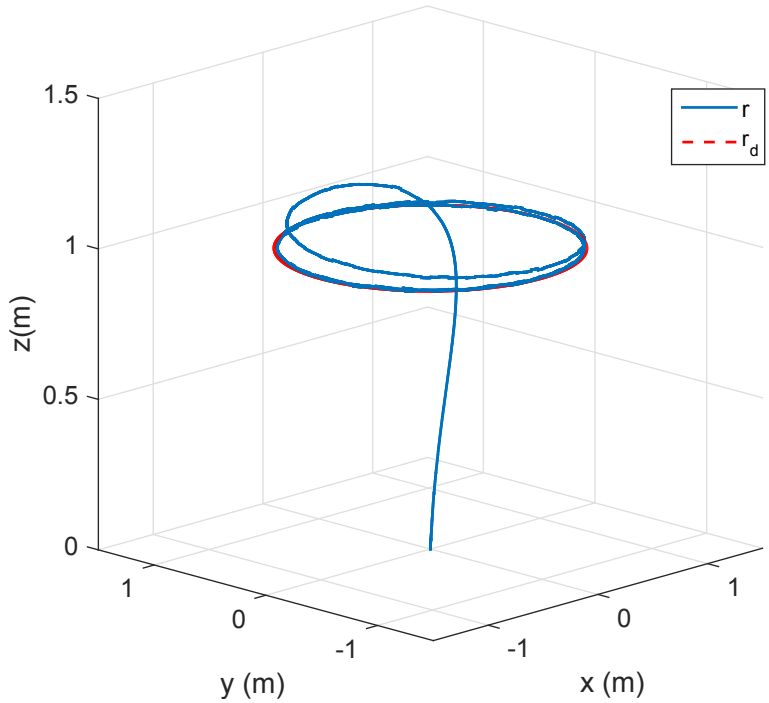


Figure 4.9: Evolution of the trajectory of quadrotor under a integral backstepping technique. The trajectory converges to the desired path.

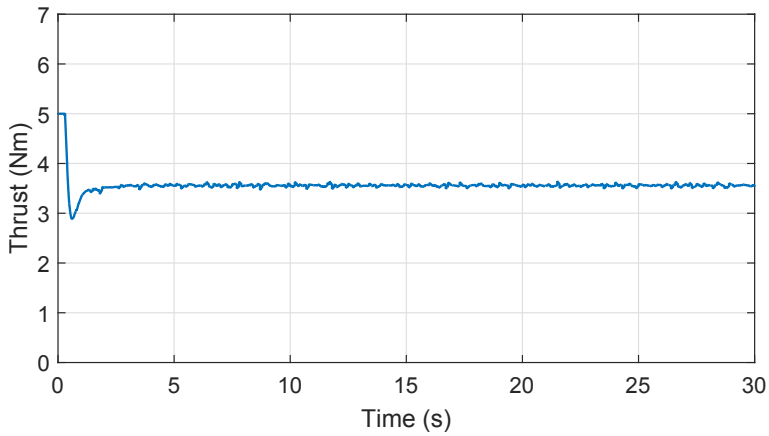


Figure 4.10: Thrust generated by the drone in the circular path.

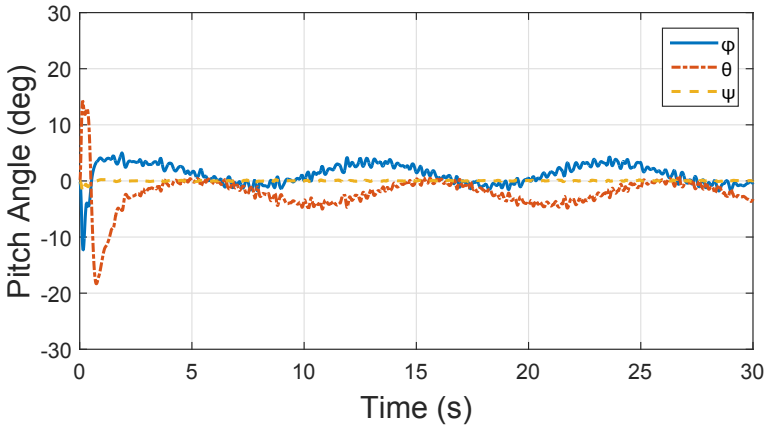


Figure 4.11: Trajectory of the Euler angles followed by the drone when doing a circular path.

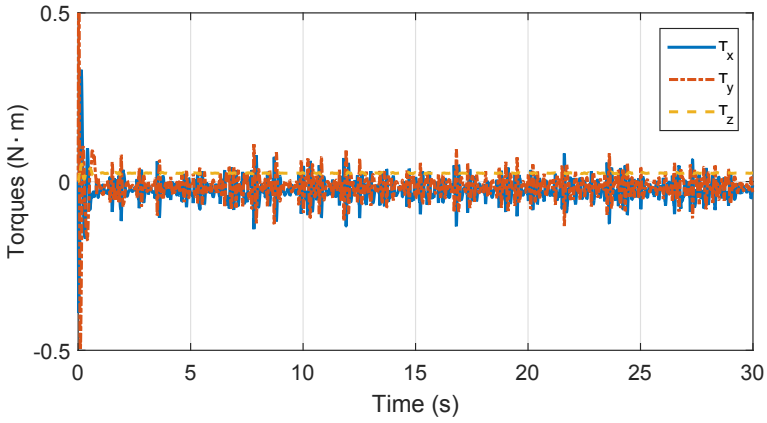


Figure 4.12: Control torques generated by the integral backstepping technique.

4.2.2 Backstepping with Adaptive Properties

This simulation tests an adaptive backstepping control which counteracts the effect of the external torques and forces added to the quadrotor system. The vehicle must follow a circular path at a rate of $0.6\,rad/s$ with a radio of $1\,m$ and a height of $1\,m$. The desired yaw angle is fixed to zero and the initial conditions for the position and the orientation are zero as well. The disturbances applied to the system are $\boldsymbol{\delta_u} = [0.125, 0.075, -0.125]N$ and $\boldsymbol{\delta_\tau} = [0.025, 0.015, -0.025]N \cdot m$. Also a white noise is added to the attitude and position in order to emulate real measures. The parameters of this simulation are shown in Table 4.4.

Table 4.4: Simulation parameters for position control of a quadrotor based on an adaptive backstepping technique.

Parameter	Value	Parameter	Value
mass	0.350 kg	Γ_1	diag(0.2, 0.2, 0.2)
$\mathbb{K}_r$	diag(2, 2, 2)	$\mathbb{K}_v$	diag(2, 2, 4)
Γ_2	diag(0.001, 0.001, 0.001)	$\mathbb{K}_\eta$	diag(4, 4, 4)
$\mathbb{K}_\omega$	diag(8, 8, 8)	$\mathbb{I}$	diag(0.008, 0.008, 0.006) kg·m^2

The results in Fig. 4.14 shows the quadrotor follows the desired trajectory even in presence of disturbances. Fig. 4.13 presents the evolution of the orientation according to the circular path. In Figs. 4.15 and 4.16, it is shown the disturbance estimations from the adaptive algorithm. These estimations are approximately the values corresponding to the external forces and torques. The torques generated by this adaptive algorithm is displayed in Fig. 4.17. Theses graphics show the robustness of the control algorithm before noisy measurements and external perturbations.

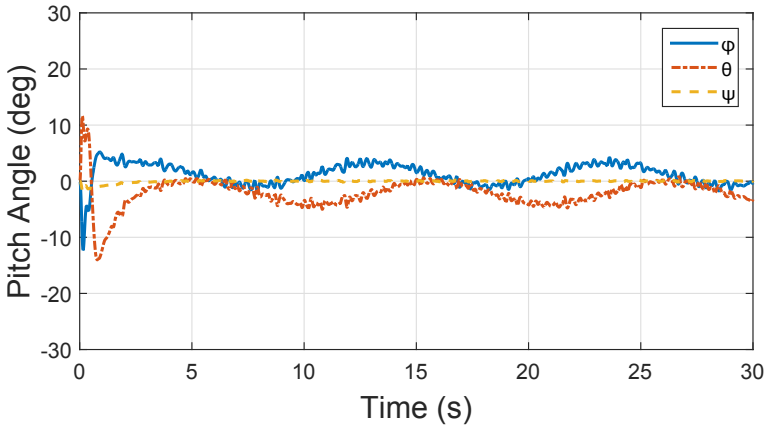


Figure 4.13: Euler angles described by quadcopter when following a circular path. The yaw angles is fixed to zero.

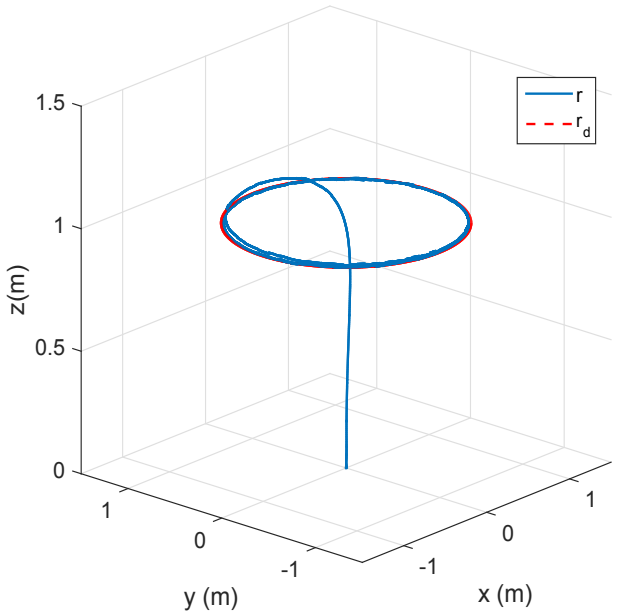


Figure 4.14: Trajectory followed by quadcopter when following an circular path in presence of constant disturbances.

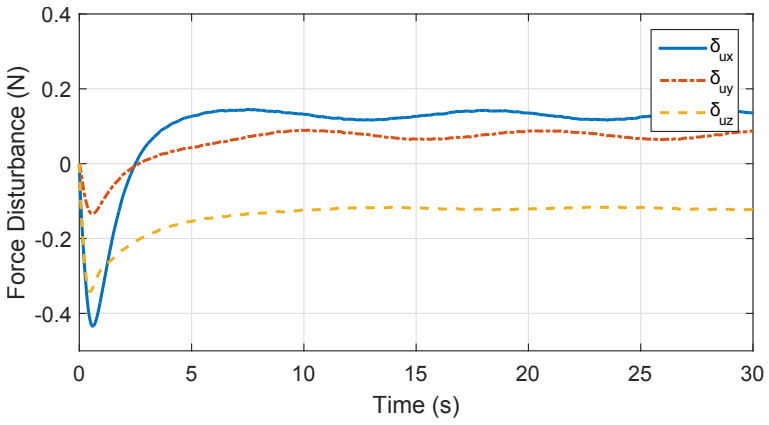


Figure 4.15: Estimation of the force disturbance applied to the quadrotor. The values are about those of the external forces applied in the position model.

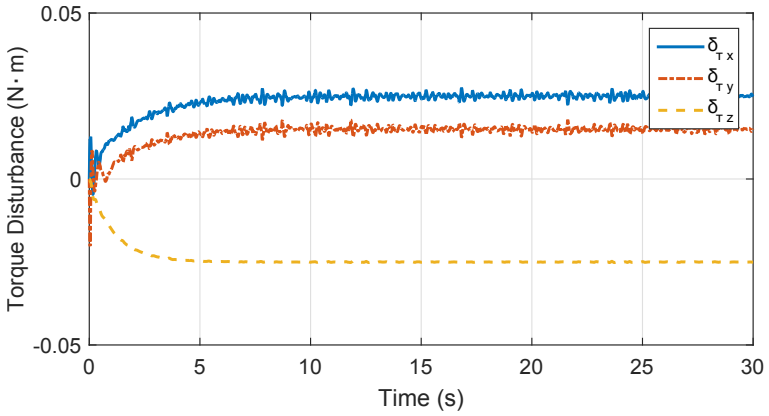


Figure 4.16: Estimation of the torque disturbance applied to the quadrotor. The values are about those of the external torques applied in the attitude model.

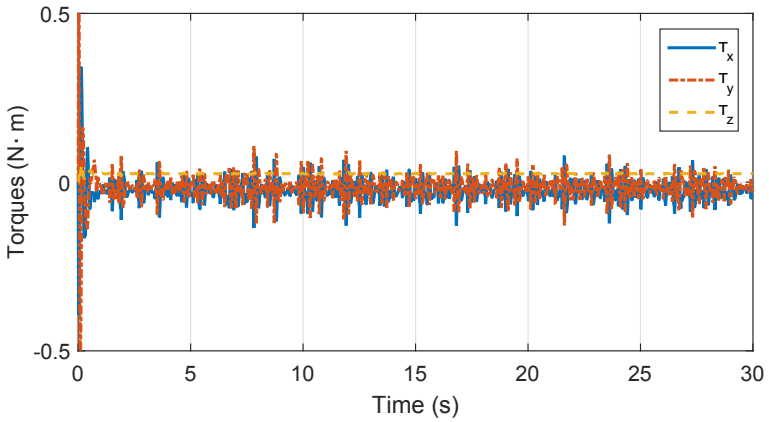


Figure 4.17: Torque generated by the adaptive algorithm. The signals are noisy because of noises added to the measures of the position and Euler angles

4.2.3 Nested Saturations

The algorithm based on nested saturations was applied to the cart and this time is extended and applied to the quadcopter. The simulation consists in following a circular trajectory at a rate of $0.6\,rad/s$ and at a height of $1\,m$. Somme disturbances are added to the model in order to evaluate the robustness of the algorithm. Theses external perturbations are $\delta_u = [0.025, 0.015, -0.025]$ and $\delta_\tau = [0.125, 0.075, -0.125]$. Also a noise is added to the measure of the position and orientation. The attitude control is carried out by the adaptive backstepping technique designed in the control algorithm section. The simulation parameters are shown in Table 4.5. According to Fig. 4.19, the vehicle follows the circular trajectory in spite of

Table 4.5: Simulation parameters for position control of a quadcopter based on a nested saturation algorithm.

Parameter	Value	Parameter	Value
mass	0.350 kg	$\mathbb{K}_1$	diag(1.5, 1.5, 1.5)
$\mathbb{M}_1$	diag(1, 1, 1)	$\mathbb{K}_p$	diag(2, 2, 2)
$\mathbb{K}_2$	diag(2, 2, 2)	$\mathbb{M}_2$	diag(0.5, 0.5, 0.5)
$\mathbb{K}_q$	diag(4, 4, 4)	$\mathbb{K}_\eta$	diag(4, 4, 4)
$\mathbb{K}_\omega$	diag(32, 32, 32)	Γ_2	diag(0.001, 0.001, 0.001)
$\mathbb{I}$	diag(0.008, 0.008, 0.006) kg·m^2		

disturbances. The thrust values are limited up to $4N$ by the saturation algorithm as shown in Fig. 4.18. The orientation of the vehicle is shown in Fig. 4.20. The estimations of the adaptable algorithm are shown in Fig. 4.21 which agree to the values of the disturbance in the attitude model. This combination of the saturation technique for the position control and the adaptive backstepping for the attitude model shows good performance during the tracking.

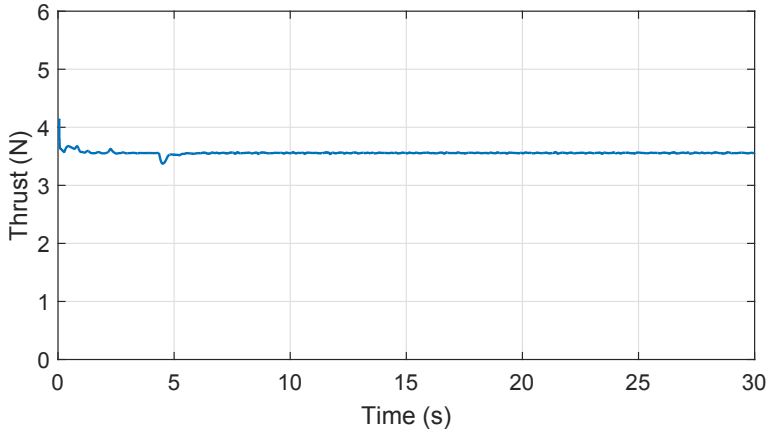


Figure 4.18: Thrust generated by the saturated control law.

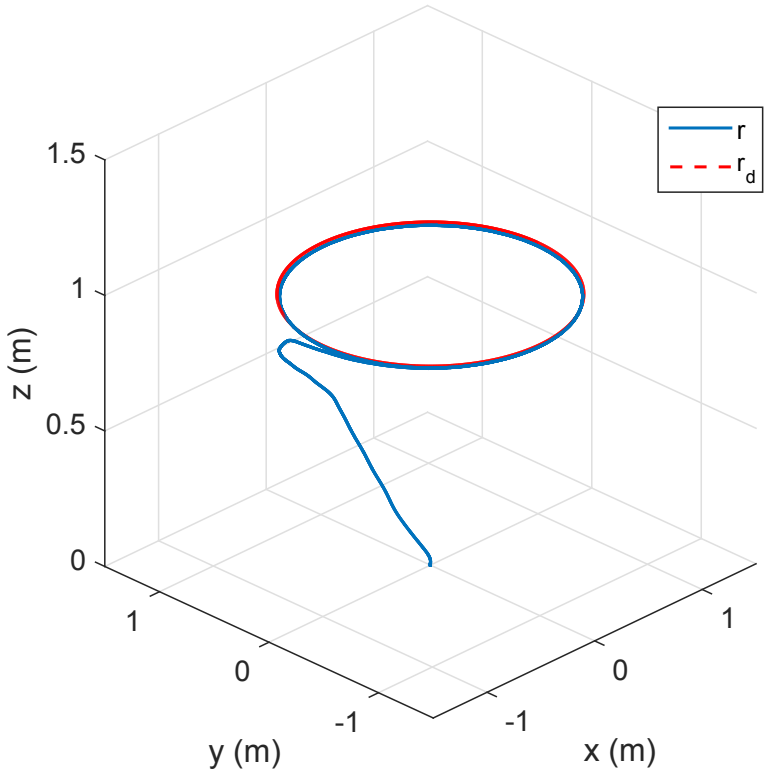


Figure 4.19: Tracking of a circular trajectory by quadcopter using an algorithm based on nested saturations. The following is carried out in presence of disturbances added to the system model.

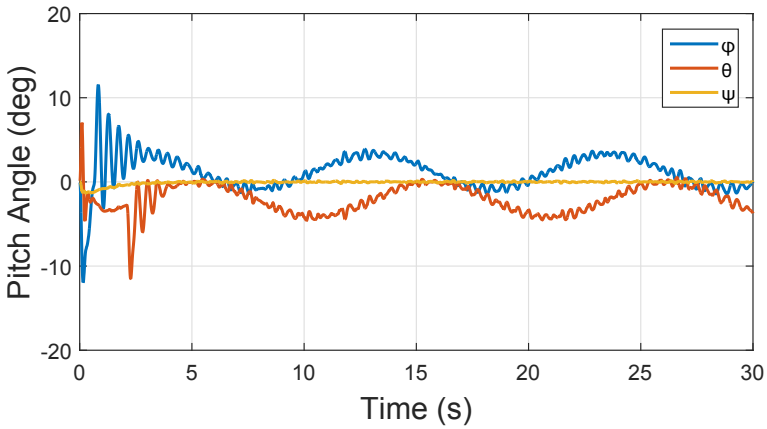


Figure 4.20: Evolution of the Euler angles during the tracking of the circular path.

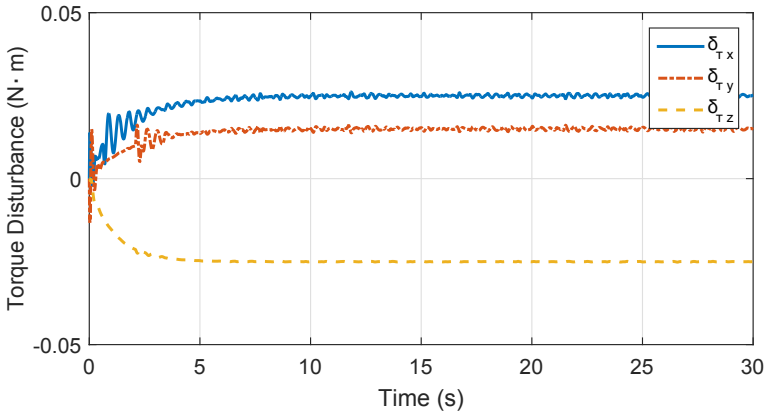


Figure 4.21: Estimation of the disturbances added to the attitude model. The algorithm uses this estimation to counteract the effect of the disturbances.

4.2.4 Nonlinear Technique

The nonlinear algorithm applied to the cart is extended to the case of a quadcopter. it was necessary to add a new factor K_v in order to counteract the lateral derivation to the target position. This term helps to stabilize the system in aerial mode and simulates the friction presented on ground mode. In this simulation, the vehicle must follow a circular trajectory at a rate of $0.6\,rad/s$ and at a height of $1\,m$. The desired yaw angle and the initial conditions are set to zero. The height control is carried out by a PID controller and the control for the xy position uses the same principle described for in cart mode. In order to test the robustness of the algorithm, disturbances in the position and in the attitude model was added. These disturbances ares $\boldsymbol{\delta_u} = [0.125, 0.075, -0.125]\,N$ and $\boldsymbol{\delta_\tau} = [0.025, 0.015, -0.025]\,N \cdot m$. Moreover, in order to emulate noisy measures in the position and the orientation, a white noise was added to them. The position control uses the nonlinear technique and the attitude control uses the adaptive backstepping technique developed in the control section. The simulation parameters are shown in the Table 4.6.

Table 4.6: Simulation parameters for position control of a quadcopter based on a nonlinear technique.

Parameter	Value	Parameter	Value
mass	0.350 kg	K_p	1
K_d	6	K_v	1
K_{pz}	2	K_{dz}	2
K_{iz}	0.1	$\mathbb{K}_\eta$	diag(4, 4, 4)
$\mathbb{K}_\omega$	diag(32, 32, 32)	Γ_2	diag(0.001, 0.001, 0.001)
$\mathbb{I}$	diag(0.008, 0.008, 0.006) kg·m^2		

The vehicle tracks the reference trajectory as can be seen in Fig. 4.22. The attitude control estimates the external torques acting as perturbations and helps to reduce their effects. Fig. 4.25 shows the estimations of the external torques which agree with the values used in simulation. The thrust in Fig. 4.23 is not constant as in the terrestrial mode. The Euler angles described by the quadcopter are displayed in Fig. 4.24. The yaw angle is zero and the roll and pitch angles vary according to the circular path.

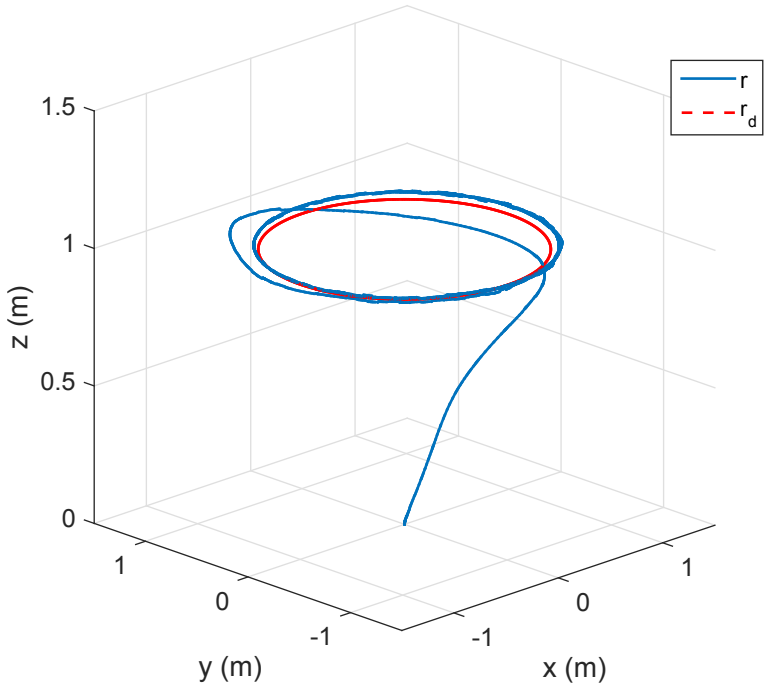


Figure 4.22: Trajectory described by a quadcopter during the following of a circular path. The nonlinear algorithm makes the quadrotor follows the desired trajectory in presence of external perturbations.

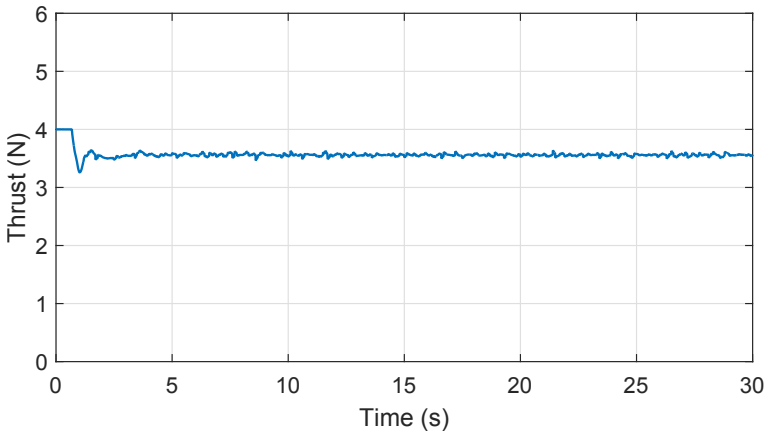


Figure 4.23: Thrust employed by the quadcopter using a nonlinear technique for position control.

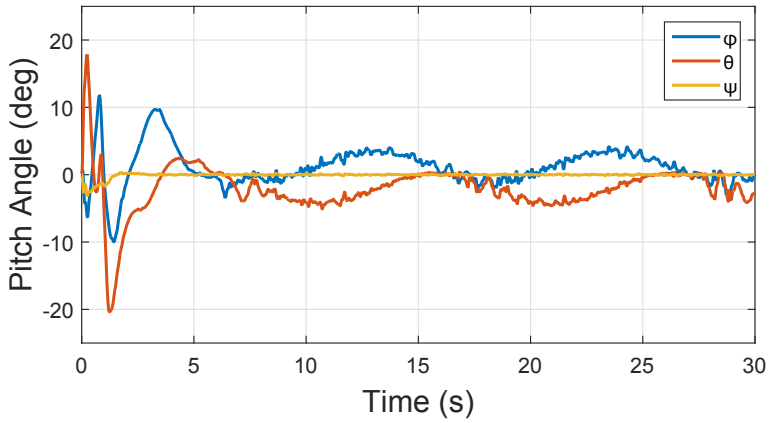


Figure 4.24: Orientation of the vehicle represented by the Euler angles of a quadrotor doing a tracking of a circular path.

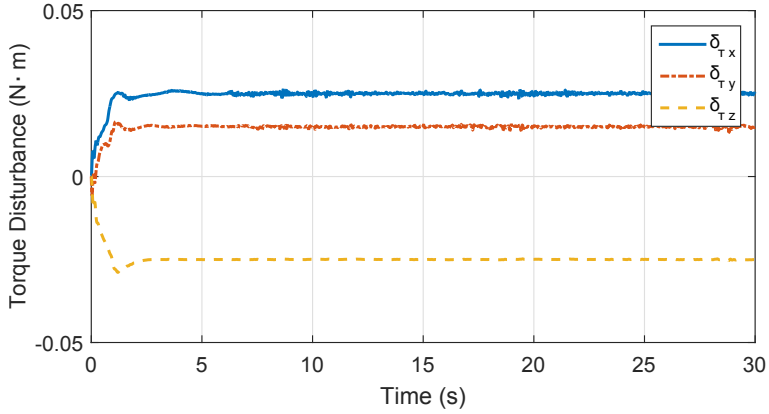


Figure 4.25: Estimations of the external torques acting as disturbances.

4.2.5 Intermediary Quaternions

The intermediary quaternion approach lets to control the quadcopter vehicle in a wide range of orientations. Also this control avoid the unwinding phenomenon observed with the classical quaternions. These properties will make the drone robust before perturbations and capable of doing complex maneuvers. These simulations validate this control algorithm based on the intermediary quaternions. The required task is to follow a circular path at a speed of $1.6\,rad/s$ and a height of $1\,m$. It is required the vehicle front follows the circular trajectory. In order to give a physical sense to the orientation, this is expressed in Euler angles obtained from the rotation matrix of the body. The initial orientation is $(\psi_0, \theta_0, \phi_0) = (180°, 0°, 0°)$ and the initial position is $(x_0, y_0, z_0) = (1, 0, 0)\,m$. The position control uses a backstepping-based algorithm and the attitude algorithm uses the intermediary quaternions. Also, it is added two disturbances $\boldsymbol{\delta_u} = [0.125, 0.075, -0.125]\,N$ and $\boldsymbol{\delta_\tau} = [0.025, 0.015, -0.025]\,N \cdot m$ and a white noise to the position and attitude measures in order to test the robustness of the algorithm. The simulation parameters are shown in the Table 4.7.

Table 4.7: Simulation parameters for position control of a quadrotor basing its attitude control on intermediary quaternions.

Parameter	Value	Parameter	Value
mass	0.260 kg	$\mathbb{I}$	diag(0.008, 0.008, 0.006) kg·m^2
$\mathbb{K}_{Ir}$	diag(0.01, 0.01, 0.01)	$\mathbb{K}_r$	diag(8, 8, 4)
$\mathbb{K}_v$	diag(4, 4, 4)	$\mathbb{K}_q$	diag(32, 32, 32)
$\mathbb{K}_{\omega e}$	diag(128, 128, 128)	$\mathbb{K}_{iq}$	diag(32, 32, 32)

In Fig. 4.26 the drone is tracking the desired path even though the presence of external perturbations. In Fig. 4.27 the error quaternion is depicted at each instant and its value is displayed around $1 + \mathbf{0}$. In Figs. 4.28 and 4.29, the evolution of the quadcopter orientation is given in Euler angles. The yaw angle varies according to the condition impose during the tracking. The vehicle takes an almost constant pitch angle during the tracking and also the centrifugal force is compensated by a small roll angle.

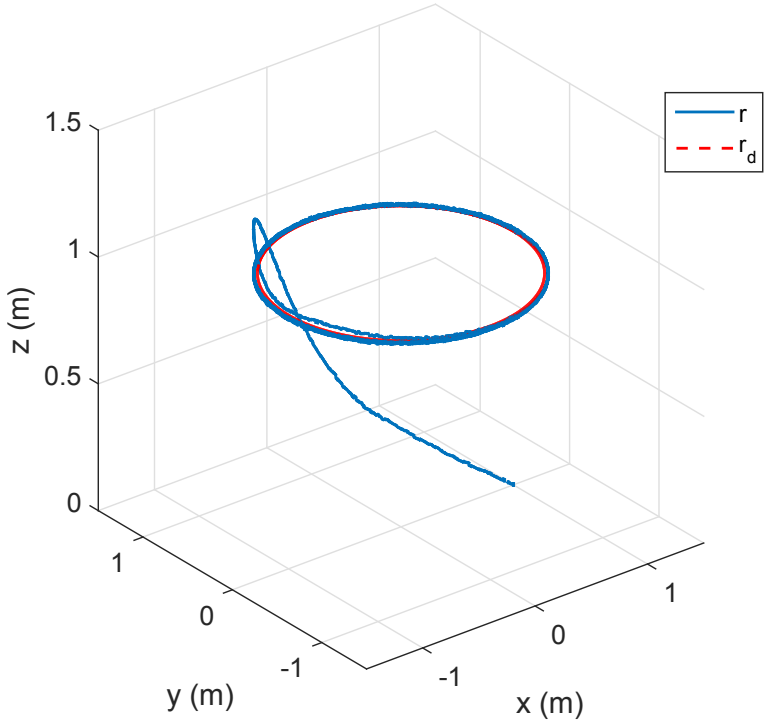


Figure 4.26: Quadcopter trajectory using intermediary quaternions in the tracking of a circular path. The drone follows the reference at a rate of $1.6\,rads/s$ in presence of disturbances.

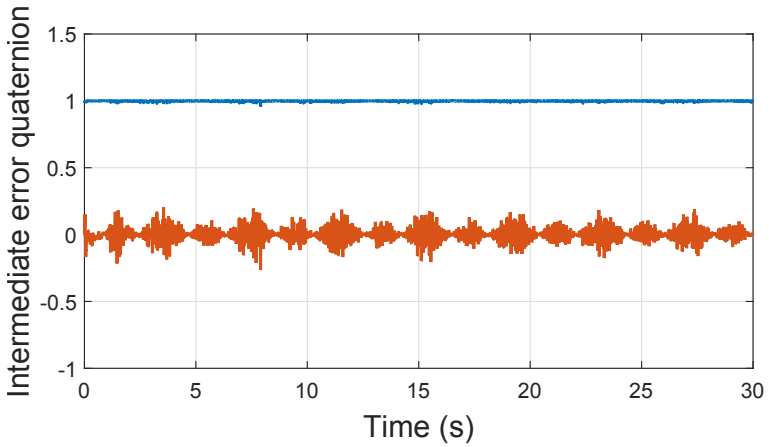


Figure 4.27: Intermediary error quaternion. This quaternion converge to the value $1+\mathbf{0}$ which proves the convergence of the attitude to the one generated by the position control algorithm.

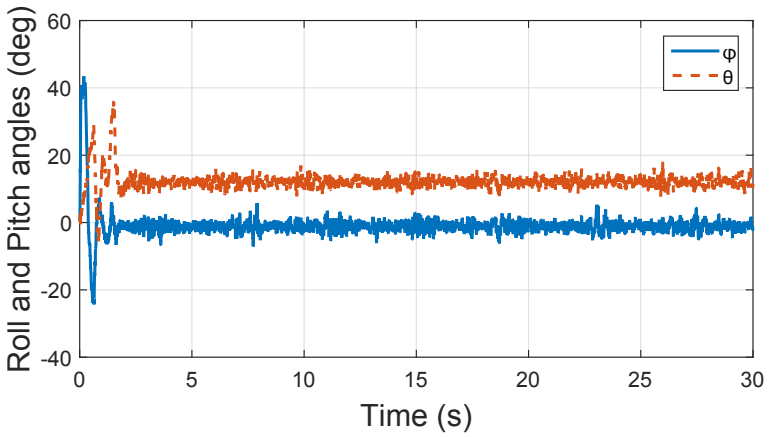


Figure 4.28: Roll and pitch angles. These angles give a better understanding of the attitude evolution during the tracking. The quadrotor vehicle is leaning forward in order to pursue the reference. The roll angle compensates the centrifugal force.

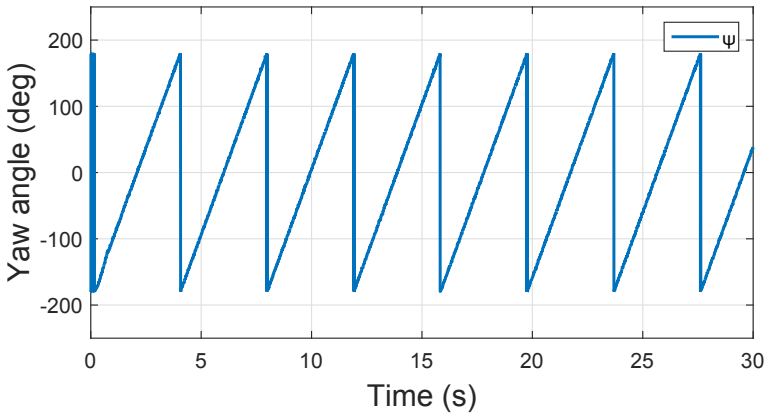


Figure 4.29: Yaw angle of the vehicle. This angle varies according to position in the circular path, the front vehicle must be tangent at each instant.

4.2.6 Application of Disturbance Observer

The disturbance observer could be used in the control loop in order to improve the performance of the control algorithm. In this simulation a backstepping position control law is employed without integral or adaptive properties. The attitude control use an adaptive backstepping algorithm which helps to compensate the disturbances in the attitude model. The task is to follow a circular trajectory at a rate of $0.6\,rad/s$ with radio of $1\,m$ and a height of $1\,m$. The disturbances used are $\boldsymbol{\delta_u} = [0.125, 0.075, -0.125]\,N$ and $\boldsymbol{\delta_\tau} = [0.025, 0.015, -0.025]\,N{\cdot}m$. The parameter used in this simulation are shown in Table 4.8.

Table 4.8: Simulation parameters for position control of a quadrotor based on an integral backstepping technique.

Parameter	Value	Parameter	Value
mass	0.350 kg	$\mathbb{I}$	diag(0.008, 0.008, 0.006) kg·m^2
$\mathbb{K}_r$	diag(2, 2, 2)	$\mathbb{K}_v$	diag(2, 2, 2)
Γ_2	diag(0.001, 0.001, 0.001)	$\mathbb{K}_\eta$	diag(4, 4, 4)
$\mathbb{K}_\omega$	diag(32, 32, 32)		

The trajectory tracking is shown in Fig. 4.30. The performance of this algorithm control could be appreciated in the Fig. 4.31. This last figure shows the norm of the position error at every instant.

Looking for improving the performance of the position tracking, a disturbance observer is implemented. The estimations coming from this observer are displayed in Fig. 4.32 which corresponds to the values used in simulation.

The idea is to use these estimations in the control loop and to reduce the tracking error. First, it is necessary to compute an approximation of the transfer function between the desired force $\boldsymbol{u_d}$ generated by the control algorithm and the output force of the quadrotor system, that is, $\boldsymbol{u} = \boldsymbol{u_{real}} - \hat{\boldsymbol{\delta}}_{\boldsymbol{u}}$. In order to simplify, each component of the output force is considered independent from the others, this leads to have a system in the form

$$\begin{aligned} U_x(s) &= G_x(s) \cdot U_{dx}(s) \\ U_y(s) &= G_y(s) \cdot U_{dy}(s) \\ U_z(s) &= G_z(s) \cdot U_{dz}(s) \end{aligned} \tag{4.1}$$

where $U_i(s)$ stands for the Laplace transform for force u_i, $G_i(s)$ means the transfer function for the component i and U_{di} indicates the component i of the reference force.

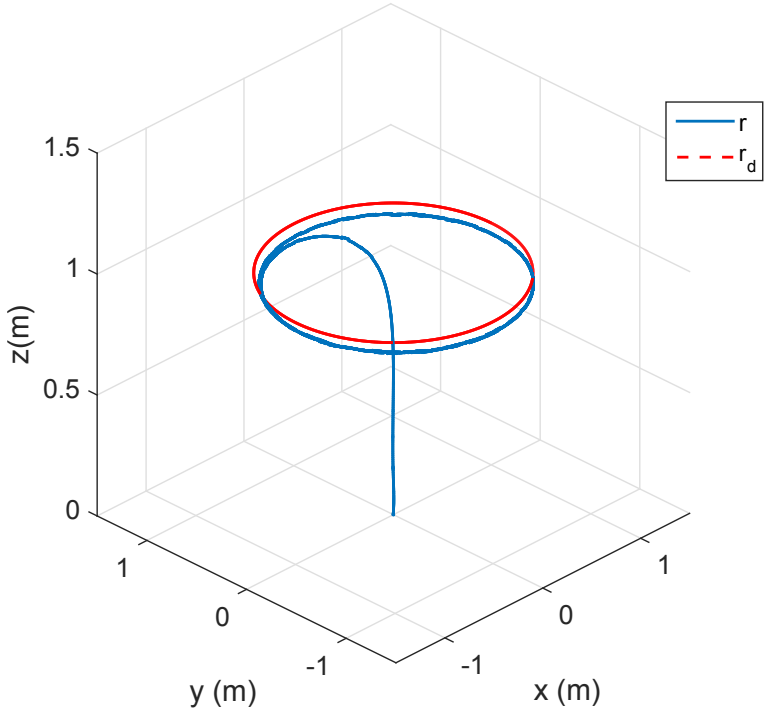


Figure 4.30: Trajectory followed by quadcopter under external perturbations.

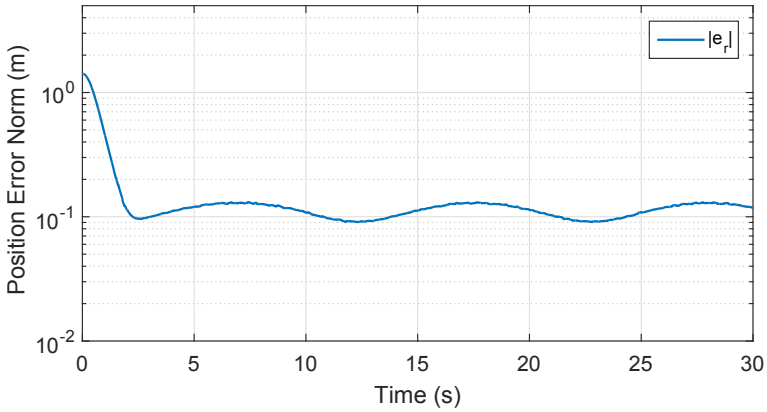


Figure 4.31: Norm of the position error during the tracking of the circular trajectory.

With the helps of the tool *ident* from Matlab®, the transfer functions are shown in (4.2).

$$\begin{aligned} G_x(s) &= \frac{-0.3205s + 0.715}{s + 0.7167} \\ G_y(s) &= \frac{-0.3247s + 0.5439}{s + 0.5448} \\ G_z(s) &= \frac{-0.3259s + 1.094}{s + 1.096} \end{aligned} \tag{4.2}$$

The next step is to invert these transfer functions. From Eqn. (4.2), these functions

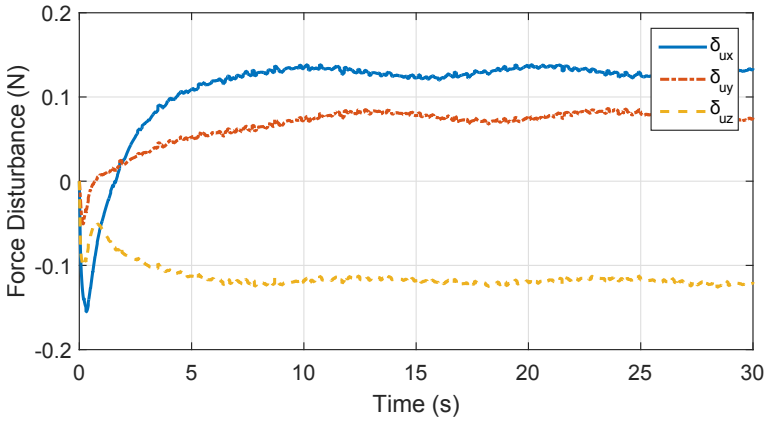


Figure 4.32: Estimation of the applied perturbations $\boldsymbol{\delta_u} = [0.125, 0.075, -0.125]$.

are of non-minimum phase and thus, the inversion will produce not stable transfer functions. Therefore, they need to be approximated by stable transfer functions and taking into account that the disturbances are constant or enough slow, the inverted transfer functions could be a low-pass filter with unitary gain. Hence, for low frequencies it holds

$$G_x(s) \approx G_y(s) \approx G_z(s) \approx \frac{1}{0.1s + 1} \tag{4.3}$$

This approximation is suitable when the estimation are noisy and the filter helps to cancel the high frequencies. In Fig. 4.34 the new trajectory fits better the desired circular path than without the compensation. Fig. 4.33 shows the position error has reduced in comparison to the control without the compensation. If the attitude control had not had the adaptive property, the estimation of this observer would include the effect of the perturbation in the orientation. Then, this observer could help to compensate also the unknown dynamics and accordingly to improve the performance.

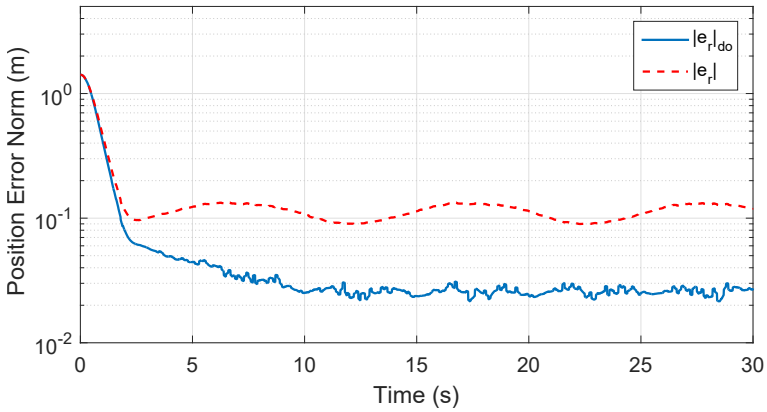


Figure 4.33: The norm of the position error with compensation using the disturbance observer (do). The new error in solid line is lower than the error without compensation in dashed line.

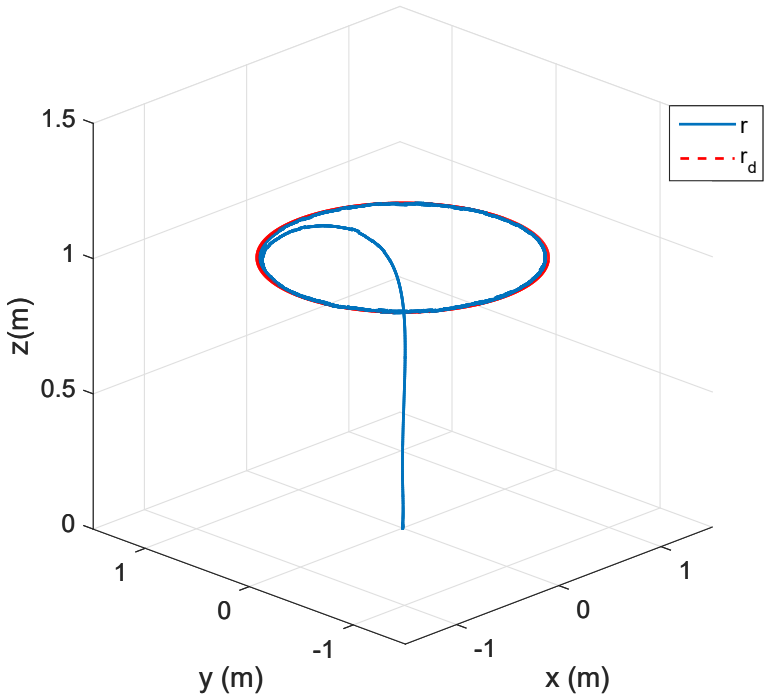


Figure 4.34: The new trajectory fits better the path reference than without compensation. The use of the disturbance error helps to reduce the gap between the two trajectories.

4.3 Hybrid Vehicle

The control strategy for the hybrid vehicle is to design an algorithm with the same structure for the terrestrial and aerial modes. Therefore, when the vehicle passes to ground mode the algorithm does not change from one technique to another, instead it changes only the parameter values of the algorithm. Similarly when the vehicle passes to a quadcopter mode, the algorithm changes to the right values for this operation mode. The transition between these two value sets could be made by the means of a set of first order differential equations. Therefore, the transition reduces to change parameters values instead of the control technique. The position control will switch its parameter values according to the operation mode and the attitude control will be based on the adaptive backstepping approach. The adaptable law is chosen in order to compensate the unknown dynamics in the attitude model and also the not modeled dynamics in our model. Fig. 4.35 describes the control strategy for the hybrid vehicle. The position control generates the thrust f and the desired control forces $\boldsymbol{u_d}$, then the attitude control produces the torques $\boldsymbol{\tau}$ which rotate the vehicle to the right orientation in order to move to the desired position.

The saturation-based algorithm is suitable because in cart mode it is able to avoid the take-off. Then, when the vehicle needs to fly, the thrust depends on the control forces generated by the algorithm for the aerial mode. Another alternative could be the nonlinear technique developed previously. This last technique generates the desired orientation the vehicle must follow in order to reach te desired position. The thrust is constant in ground mode but in

aerial mode it depends on a PID algorithm for controlling it altitude.

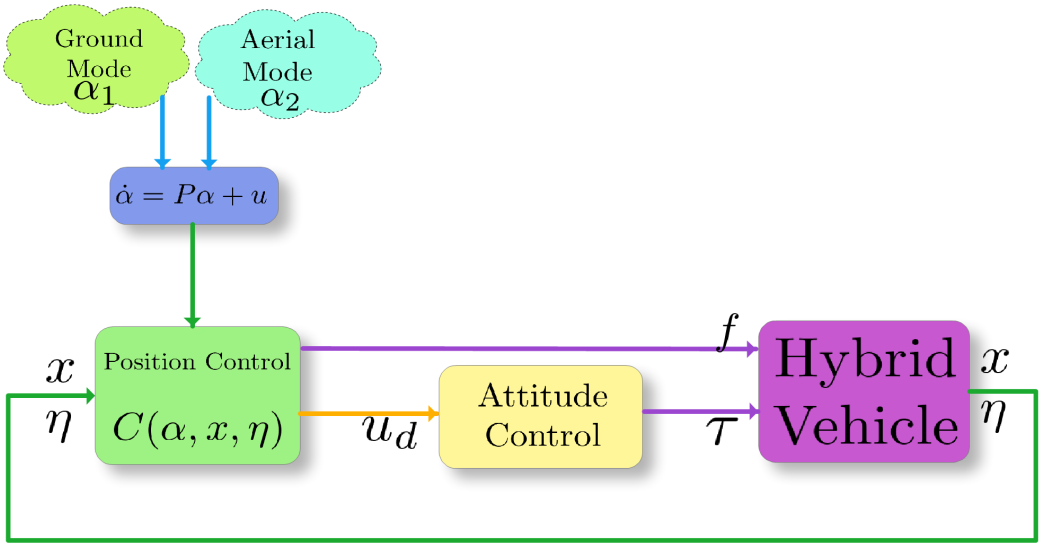


Figure 4.35: Control strategy for a hybrid terrestrial and aerial vehicle. The control law has the same structure for the two operations modes, there is necesssary to change from one set of parameter values to another when the vehicle changes its operation mode.

4.3.1 Nested Saturations

In this simulation the vehicles must follow a hybrid terrestrial and aerial trajectory. The trajectory is composed by four paths, the first one is the take-off and in this phase the vehicle has the orientation $(\psi_0, \theta_0, \phi_0) = (90°, 0, 0)$ and initial position $(x_0, y_0, z_0) = (0.5, 0, 0)\,m$. Then, the vehicle moves to the position $(x_1, y_1, z_1) = (0.5, 0.3, 1)\,m$ and the second phase begins. During this phase, the vehicle moves in a half circular path with a radio of $0.5\,m$ and at a height of $1\,m$. This phase ends at the point $(x_2, y_2, z_2) = (-0.5, 0.3, 1)\,m$. The third phase is the landing until reaching the position $(x_3, y_3, z_3) = (-0.5, 0, 0)\,m$. Then, the vehicle changes from aerial to ground mode. The forth phase is a half circular path over the ground with a radio of $0.5\,m$. The final position of this phase is the initial position of the hybrid trajectory. In addition, the take-off and landing must be done in $3\,s$ and the circular paths covered at a rate of $0.2\,rad/s$. Also, it is considered the presence of disturbances onto the vehicle, theirs values are $\boldsymbol{\delta_u} = [0.05, 0.03, 0]\,N$ and $\boldsymbol{\delta_\tau} = [0.025, 0.015, -0.025]\,N \cdot m$. Furthermore, a white noise is added to the position and attitude measure in order to emulate real scenarios. The parameters of this simulation are shown in the Table 4.9

From Fig. 4.36, the vehicle follows the desired hybrid trajectory and the robustness of the algorithm is proved since the following is made in presence of disturbances. In Fig. 4.37 it is observed the changes in the thrust depending on the vehicle mode. The vehicle consumes less energy in ground mode, this would be advantageous when the energy is an important fact to take care. Also, the pitch angle is bigger in ground mode because the thrust is smaller than the aerial mode, this is observed in Fig. 4.38. The sign of the pitch angle will depend on yaw

Table 4.9: Simulation parameters for position control of a hybrid vehicle based on a nested saturation algorithm. The subscript a means aerial and the g stands for ground. The attitude control parameters are equal for the two operation modes.

Parameter	Value	Parameter	Value
mass	0.350 kg	$\mathbb{K}_{1a}$	diag(1.5, 1.5, 1.5)
$\mathbb{M}_{1a}$	diag(1, 1, 1)	$\mathbb{K}_{pa}$	diag(1, 1, 1)
$\mathbb{K}_{2a}$	diag(1, 1, 2)	$\mathbb{M}_{2a}$	diag(0.5, 0.5, 0.25)
$\mathbb{K}_{qa}$	diag(4, 4, 32)	$\mathbb{K}_{\eta}$	diag(1, 1, 1)
$\mathbb{K}_{\omega}$	diag(32, 32, 32)	Γ_2	diag(0.001, 0.001, 0.001)
$\mathbb{I}$	diag(0.008, 0.008, 0.006) kg·m^2	K_{fg}	0.5
K_{1g}	diag(1, 1)	M_{1g}	diag(1, 1)
K_{2g}	diag(1, 1)	M_{2g}	diag(0.5, 0.5)
K_{pg}	diag(1, 1)	K_{qg}	diag(3, 3)
ρ_g	2	$K_{g\,min}$	0.05

angle the vehicle takes during the tracking. The yaw angle is presented in Fig. 4.39. This angle is constant during the take-off and the landing phases. In the other phases, the vehicle's front is in a tangent direction to the followed path.

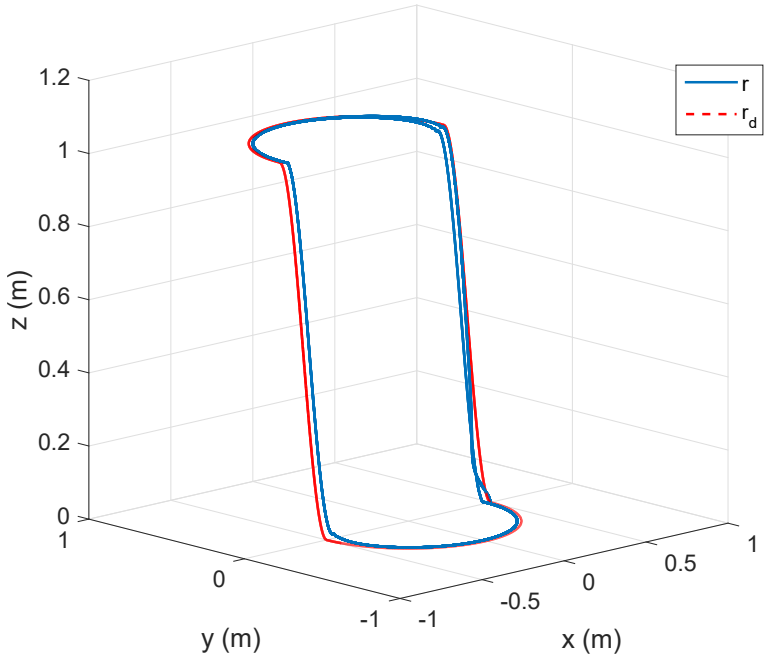


Figure 4.36: Hybrid trajectory described by vehicle using nested saturations. The vehicle switches between the parameter values for each operation mode.

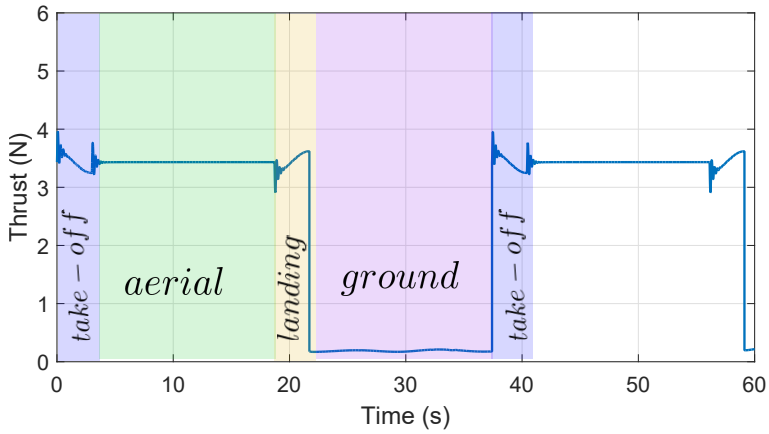


Figure 4.37: Thrust required by hybrid vehicle when doing a following of a terrestrial and aerial path.

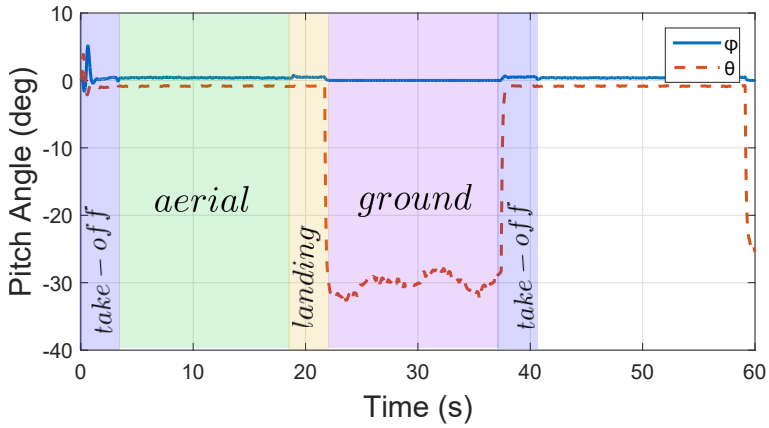


Figure 4.38: Evolution of the roll and pitch angles during the following of the hybrid trajectory.

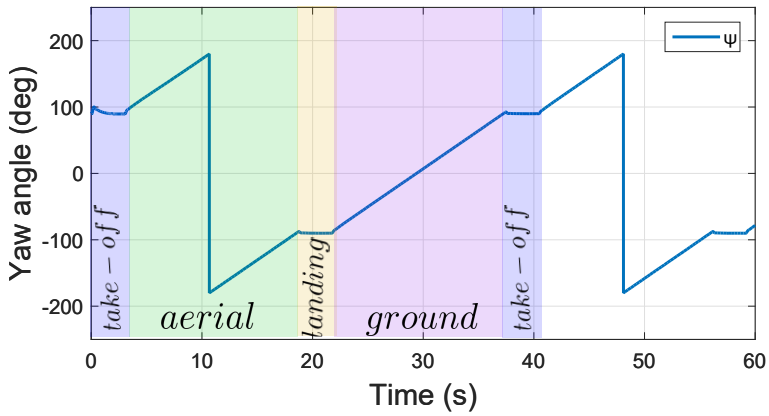


Figure 4.39: Orientation of the vehicle while following a hybrid trajectory. The yaw angle is set to be ±90 when taking-off or landing. In the circular paths, the vehicle front is tangent to the followed trajectory.

4.3.2 Non-linear Technique

This simulation tests the nonlinear control developed previously. Disturbances are added to the model in order to test the robustness and also it is considered noisy measures in the position and attitude in order to emulate real situations. The disturbances are $\boldsymbol{\delta_u} = [0.05, 0.03, -0.05]\, N$ and $\boldsymbol{\delta_\tau} = [0.025, 0.015, -0.025]\, N \cdot m$. The vehicle must follow a hybrid terrestrial and aerial trajectory changing from ground to aerial mode and vice-versa. The trajectory is composed by four paths, the first one is the take-off and in this phase the vehicle has the orientation $(\psi_0, \theta_0, \phi_0) = (90°, 0, 0)$ and initial position $(x_0, y_0, z_0) = (0.5, 0, 0)\, m$. Then, the vehicle moves to the position $(x_1, y_1, z_1) = (0.5, 0.3, 1)\, m$ and the second phase begins. During this phase, the vehicle moves in a half circular path with a radio of $0.5\, m$ and at a height of $1\, m$. This phase ends at the point $(x_2, y_2, z_2) = (-0.5, 0.3, 1)\, m$. The third phase is the landing until reaching the position $(x_3, y_3, z_3) = (-0.5, 0, 0)\, m$. Then, the vehicle changes from aerial to ground mode. The forth phase is a half circular path over the ground with a radio of $0.5\, m$. The final position of this phase is the initial position of the hybrid trajectory. In addition, the take-off and landing path must be done in $3\, s$ and the circular paths covered at a rate of $0.2\, rad/s$. The parameters of this simulation are shown in the Table 4.10

Table 4.10: Simulation parameters for position control of a hybrid vehicle based on a nonlinear control algorithm. The subscript a stands for the aerial parameters and the g means ground parameter.

Parameter	Value	Parameter	Value
mass	0.350 kg	K_{pa}	1
K_{da}	7	K_{va}	1
K_{pza}	2	K_{dza}	2
K_{iza}	0.1	$\mathbb{K}_\eta$	diag(1, 1, 1)
$\mathbb{K}_\omega$	diag(32, 32, 32)	Γ_2	diag(0.001, 0.001, 0.001)
K_{pg}	1	K_{dg}	6
f_g	3 N	$\mathbb{I}$	diag(0.008, 0.008, 0.006) kg·m^2

In Fig. 4.40, the hybrid vehicle follows the desired path and thereby it validates the nonlinear control developed before. This control algorithm compensates the disturbances added to the model. The thrust changes according the two operation modes as shown in Fig. 4.41. The vehicle follows the path in a counter-clock sense and this makes the pitch angle to be positive. The behavior of the orientation is displayed in Fig. 4.42. When the vehicle is in ground mode, the pitch angle is almost equal to the aerial mode because the thrust is set to $3\, N$, that is a little below the values required to take-off. The evolution of the yaw angle is in a way the vehicle front is always following the hybrid trajectory, as seen in Fig. 4.43.

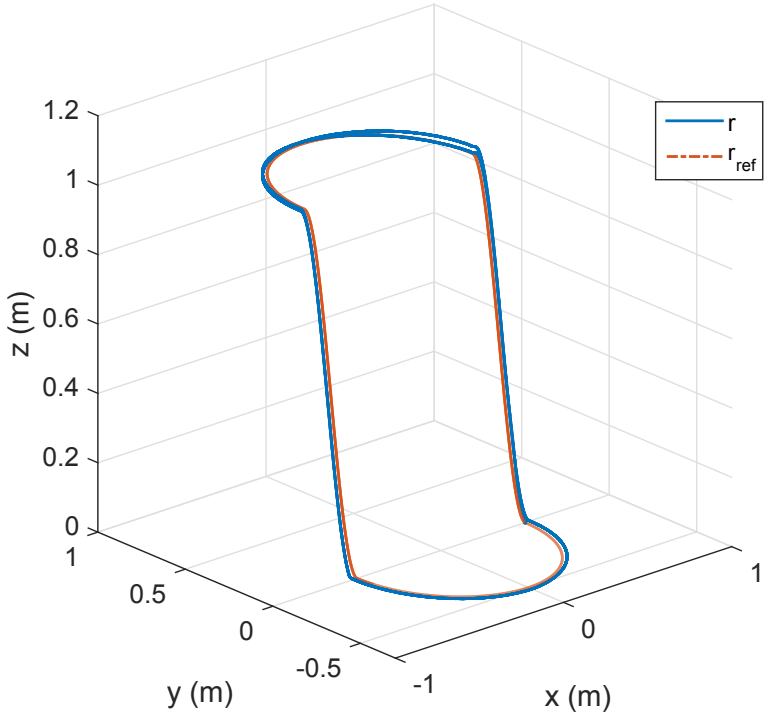


Figure 4.40: Hybrid trajectory described by vehicle using a the nonlinear technique. The vehicle switches between the parameter values for each operation mode during the transitions.

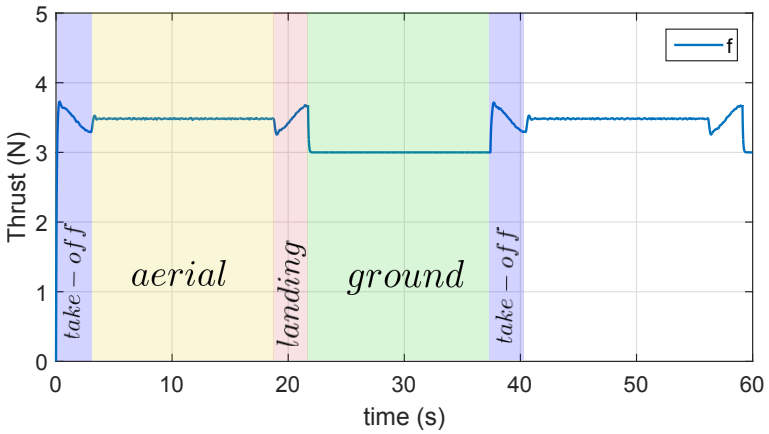


Figure 4.41: Thrust of hybrid vehicle while following a terrestrial and aerial path.

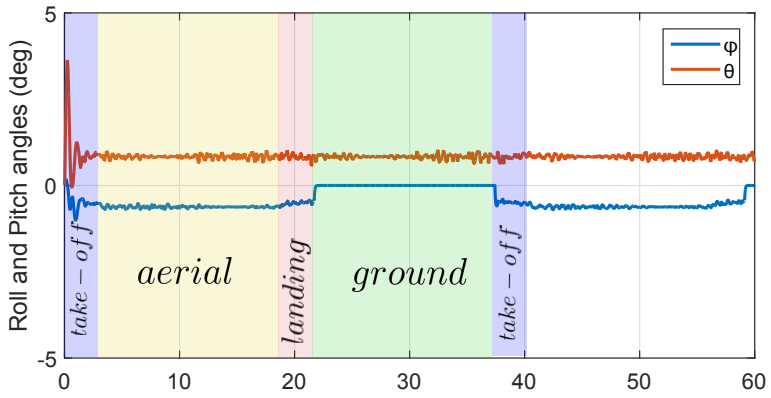


Figure 4.42: Evolution of the roll and pitch angles during the following of the hybrid trajectory.

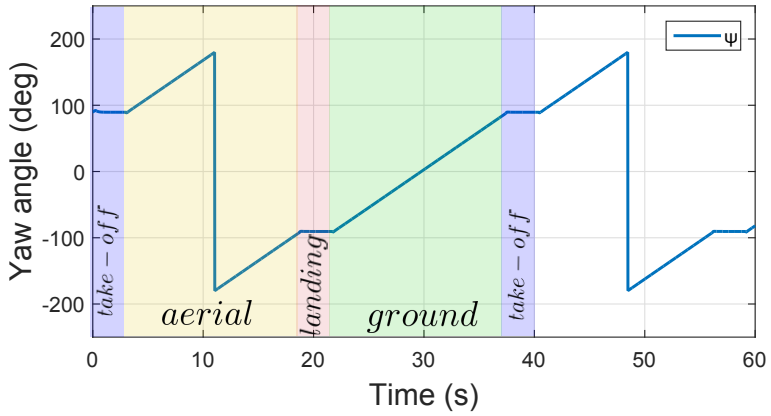


Figure 4.43: Yaw angle of the vehicle while following a hybrid trajectory. The yaw angle is set to be ± 90 when taking-off or landing and the vehicle front is tangent to the circular trajectories.

4.4 Summary

The results of these numerical simulations validated the control algorithm for the terrestrial and aerial modes. The presence of external disturbances and noise were added to the model as a way to test the robustness of the algorithms. In ground mode, the saturation and the non linear algorithm shown the vehicle followed the proposed circular trajectory with a small gap between the desired path and the described one by the vehicle. In aerial mode, several techniques were validated, the backstepping technique in two variants, that is, with integral properties and with adaptive ones. Also, the combination of nested saturations for the position control and adaptive backstepping control for the orientation part exhibited the vehicle followed the reference trajectory. After, the attitude control based in the intermediary quaternions was tested in conjunction with a backstepping algorithm for the position control part. This intermediary quaternion based control solved the issues remaining with the classical ones. Furthermore, it was corroborated a high order observer could be used as a compensation in order to improve the position control of a quadcopter. In the last subsection, we put together the previous mentioned techniques in order to develop a way to control the hybrid vehicle without changing the structure of the control algorithm. In particular, the saturation and the non linear techniques was used to control the vehicle during the tracking of an aerial and terrestrial trajectory. The control law does change its form but only the parameters values needed for the tunning of the algorithm. The goal of testing several control laws was to give an insight of the algorithm that could be better adapted to our vehicle in the experimental tests.

Chapter 5

Real-time Results

Contents

5.1 MOCA room and ground station

In order to test the developed algorithms it is necessary to know the attitude and position of the system in real time, for this, GIPSA-lab has the MOCA (motion capture).

5.1.1 MOCA room

The motion acquisition is made through infrared cameras with emitters and receivers of infrared light and also through reflecting markers attached to the moving objects or individuals.

This room is composed of 12 VICON© cameras (T40 series), attached to a metal structure in high and pointing their vision towards a common area. There are also 8 digital cameras pointing to the same area, but these ones are used for objects reconstruction or motion capture by image processing. With this system it is possible to compute the position and attitude up to 500Hz, Fig. 5.1 shows an image of the MOCA room [59].

Figure 5.1: MOCA room.

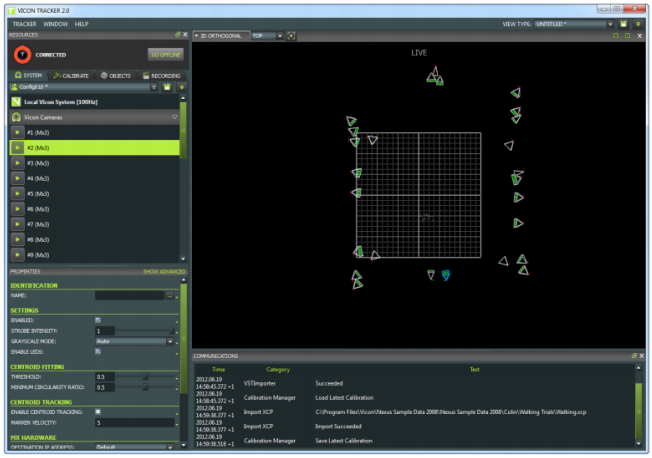
A VICON camera is an infrared camera, which emits and receives infrared rays. A set of cameras pointing towards a common area is able to detect a reflective marker. The markers are little balls of retro-reflecting materials going from 0.5 to 2cm. of diameter. The cameras emit a very special light which makes the receivers sensitive only to this one, when a marker is placed in the area covered by the cameras, it creates a single point in the plane of each one of the cameras (if the area is well covered). Then, the information is collected in a computer running the VICON© tracker software. Fig. 5.2 shows an image of the used VICON cameras and the VICON tracker environment.

5.1.2 Ground station

The ground station is composed by two computers: the first one is under the real time MATLAB/Simulink© environment and a PC target, which is under the xPC target toolbox, as well as a radio-frequency emitter.

The estimated states (attitude and position) are sent to MATLAB/Simulink through a UDP frame every 2ms, from these data, the position control algorithm is computed and implemented in real-time at 100Hz on the target PC, which uses the xPC target toolbox. xPC Target also manages communications between the host and target PC, as well as the different inputs/outputs of the real-time application.

(a)



(b)

Figure 5.2: (a)VICON cameras and (b)VICON tracker environment

The control variables are finally sent back to the system through a GIPSA-lab's built-in bridge that converts UDP frames to DSM2 protocol, for this, the radio-frequency emitter is used. Fig. 5.3 shows an overview of the computing process.

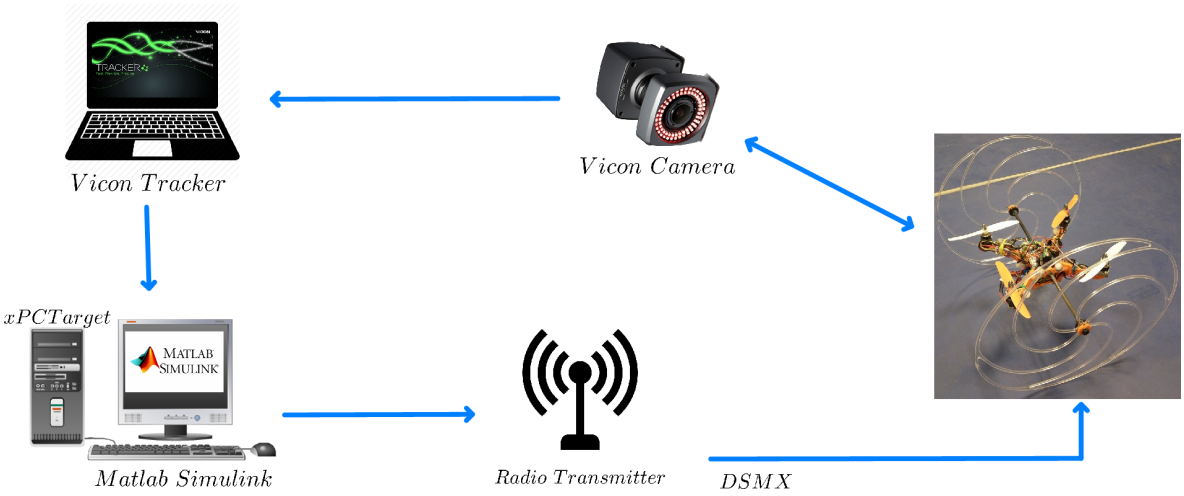


Figure 5.3: Control system process at MOCA room.

5.2 Prototypes

In the developing of this work, several prototypes were built and used during the tests in the MOCA room. At the beginning, a commercial small drone was used to test the control algorithms. This drone received the references for the orientation and the thrust, then the drone follows these references in order to move to the desired trajectory. One limitation, among others, with this prototype was the orientation control algorithm was closed and therefore it can not be adapted.

The second prototype was based on the Flexbot project. This project lets to change the orientation algorithm and, as a consequence, let us to adapt this algorithm according to the needs of this work. This protoype had some limitations and the most remarkable was the battery life during the flights. It is worth to mention the drone structure can be printed in the laboratory in a 3D printer. This allows to change the form of the structure of the drone. After, with the addition of wheels, the motors presented some difficulties to take off and as a consequence the flight duration get reduced significantly.

Posteriorly, a prototype according to our needs was conceived. The electronic card used was the Crius Multiwii AIO pro, and brushless motors were chosen because of their power and resistance to the crashes. The structure was designed and printed in the laboratory. These stages let us to learn and to perfect every next prototype.

5.2.1 NanoQX

This drone is very stable and has a smooth movement during the flight tests. Fig. 5.4 shows the concerned drone. This small drone is resistant to crash as well because of its little size its payload is very limited, about 10 gr at most. Its principal characteristics are: its weight is 20 gr., the battery used is a 1S 3.7V 150mAh 25C Li-Po Battery and it uses the DSM2 or DSMX

radio protocol. The noise produced by its helices are very low and it has a SAFE function that let it to hover and to move smoothly.

Figure 5.4: Drone NanoQx

This prototype was used to test position control laws, like nested saturations, PID controller and backstepping techniques. The results were satisfactory but the discharge of its small battery played a crucial roll during the experiments. The drone had markers in order to use the Vicon System for obtaining its orientation and position. At full battery the drone behaves as desired but after a little while the battery discharge affected the control laws performance.

This drone was transformed in a little cart after adding two light wheels as shown in Fig. 5.5. In this mode, the motors did not produce the sufficient torque to change the direction of its movement over the ground. This led to move to another prototype. For more details, see the page web of the manufacturer in [60]

5.2.2 Flexbot

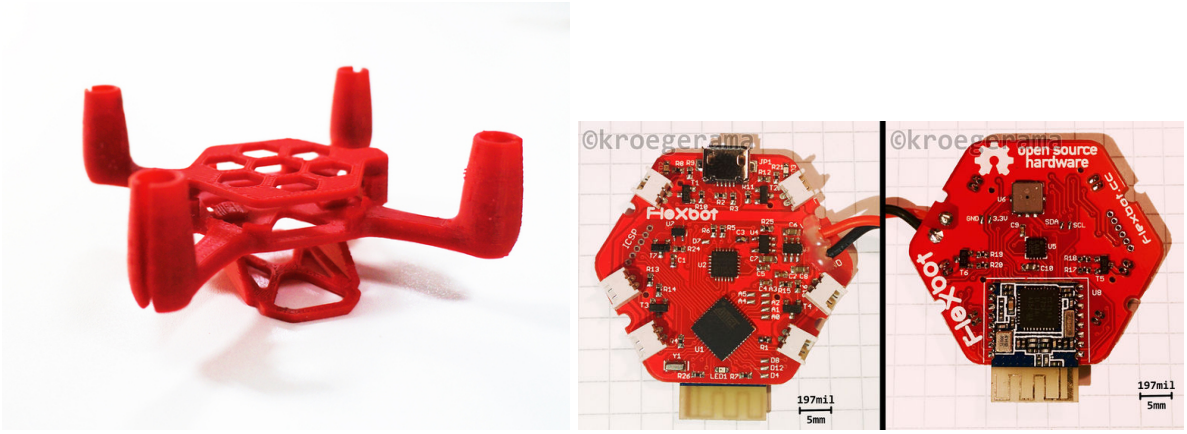
The next prototype was based on the flexbot platform. The Flexbot lets to modify the orientation code in its card and this let us to test our control algorithms. Figure 5.6 shows the flexbot hexarotor and quadrotor platforms. The flight controller board is composed by an ATMega32u4 processor, an IMU sensor (MPU6050), a magnetometer (HMCL5883L) and a barometer (BMP085). The motors are directly connected to the card and some transistors are mounted to run the DC motors. The board can be programmed with arduino, and the default firmware is based on MultiWii. A bluetooth low energy module is also mounted and communicate through a serial bus with the processor. Figure 5.7b shows the flight controller board.

Figure 5.5: NanoQx in Cart Mode

Figure 5.6: Flexbot platform, hexarotor and quadrotor.

An application for smartphone is provided to control the multicopter with bluetooth protocol. The data received by the flight controller board is organized with *MultiWii Serial Protocol* (MSP). In order to communicate between the ground station and the platform, a specific board had to be developed. It consists of an arduino MEGA, an Ethernet shield and a bluetooth shield. The arduino board is connected to the same local network as the ground station which packs the commands to fulfill the MSP and sends them through UDP to the IP adress of the arduino board at a specific port. The arduino board is running an UDP server which listen to the communication, pair the bluetooth device and relay the data received by UDP to the flexbot's bluetooth module.

This solution has been found to be reliable but was only possible because the bluetooth module of the flexbot and the arduino shield were both the same. However, flexbot changed the bluetooth module after the first batch, the bluetooth shield was then not able to be paired with the flight controller board. To overcome this issue, a *raspberry pi* and a *bluetooth dongle*



(a) (b)

Figure 5.7: 3D printed frame of the flexbot quadrotor and the flight controller board. (a) 3D printed frame. (b) Flight controller board.

have been used. The communication protocol had to be reverse engined, a python script has been developed to run an UDP server, and to send the data through bluetooth protocol.

For my work, I attached two wheels to a hexa-rotor as shown in Fig. 5.8. This new prototype let to make some tests over the ground. The experiments was mainly to generate the convenient Euler angles and thrust that could lead the vehicle to the desired trajectory and to avoid the take-off in the cart mode.

Figure 5.8: Hybrid vehicle using a Hexa-Flexbot.

The limitations of this prototype was found in flight, because the CD motors of the Flexbot was not enough powerful to hold the cart in the air. These motors get worn out very soon and also the battery duration was about 2 min minutes as most.

5.2.3 Hybrid Vehicle

The next prototype born from the need to have a vehicle that flies longer than the previous ones and to be more durable, with a structure resistant to possible crashes. The card Multiwii Crius AIO Pro was selected because this uses a similar code to the Flexbot platform and it lets us to program the orientation algorithm in the card. The used radio receiver is a DSMX compatible one and the motors are brushless. The used batteries are LiPo 2S 7.4 V 1300mAh EC2 20C E-FLITE. The new flight duration was about 10 minutes and the structure more resistant to eventually crashes during the experiments. Fig. 5.9 shows the concerned prototype. The weight of the vehicle was 260 grs. and the dimensions 30x30 cm with propellers of 5 in.

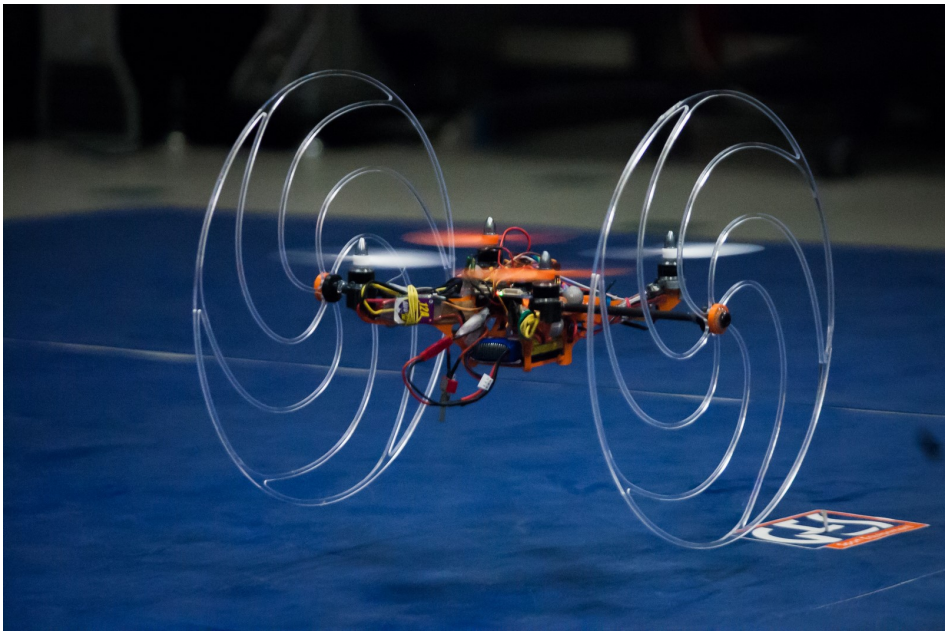


Figure 5.9: Hybrid vehicle using a Multiwii Crius Card.

Subsequently, the design of the wheels were improved according to their resistance observed in the MOCA room. These wheels were designed and made in the GIPSA Lab. The hole structure of the vehicle was also designed in GIPSa Lab, Fig. 5.10 shows a design example.

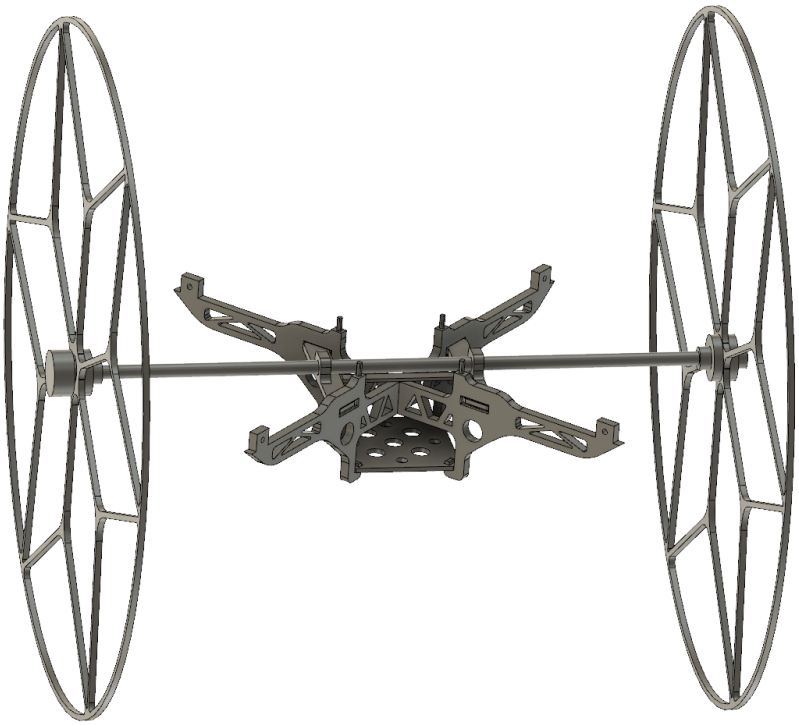


Figure 5.10: Design of the hybrid vehicle structure.

5.2.3.1 Flight Controller Card

The attitude control law for the vehicle was programmed in a Copter board, which has an IMU sensor (MPU6050) integrated by gyros and accelerometers, the HMC5883L 3-axis digital magnetometer, a MS5611-01BA03 high precision altimeter and the ATMega 2560-16AU as processor. The processor consists on a high-performance, low-power Atmel 8-bit AVR RISC-based microcontroller and it combines 256KB ISP flash memory, 8KB SRAM, 4KB EEPROM, 86 general purpose I/O lines, 32 general purpose working registers, real-time counter, six flexible timer/counters with compare modes, PWM, 4 USARTs, byte oriented 2-wire serial interface, 16-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device achieves a throughput of 16 MIPS at 16 MHz and operates between 4.5-5.5 volts.
The board dimensions are of $50 \times 50 \times 11.66$ mm and the weight of 14.5 gr. The motors are connected to the card though the speed controllers (ESC) since these are brushless motors. Fig. 5.11 shows the CRIUS flight controller board.

The firmware is based on Multiwii, however the code has been written to run on numerous platforms and flight systems. In order to edit the code, first it is necessary to specify which type of multirotor is used, for this, two software packages are needed:

- **Arduino**: the development environment which allows to edit and upload the code;
- **Multiwii**: It includes both, the open source code and the graphic user interface (GUI),

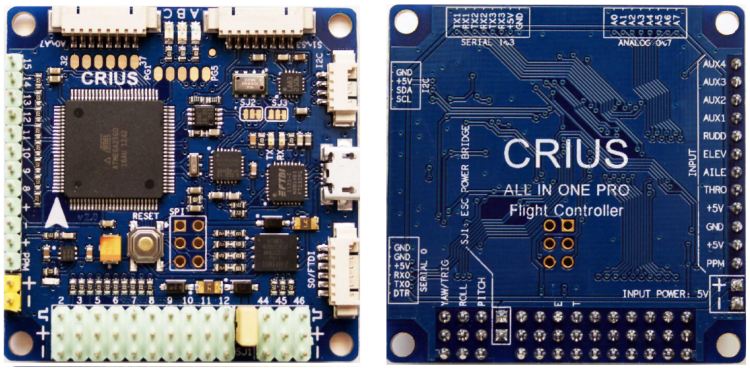


Figure 5.11: CRIUS flight controller board.

necessary for the configuration of the board parameters

5.2.3.2 Multiwii GUI

To change the card's flight parameters we need the MultiWii graphical interface, which is activated running the file MULTIWIIConf.exe. Once the board is connected to a COM port of the computer, it is possible to change the parameters of the different sensors in the board (accelerometers, gyros, magnetometer, altimeter). To better describe the functions of the GUI, here some of its modules are presented :

Sensors

The measurements of accelerometer, gyrometer, magnetometer and barometer are displayed in real time and reported on the graph, see Fig. 5.12. By clicking on the colored rectangles you can enable/disable the graph relating to a specific figure. In the rectangle in the upper right of the graph one can vary the zoom: click and hold the mouse button and drag it until get the desired value.

Flight mode

This section allocated for the remote control, switches the various flight modes by turning on or off the squares associated to the three switch positions: LOW, MID, HIGH.

- ACRO: Is the default mode when HORIZON or ANGLE are not activated.
- ANGLE: The multicopter performs a standard fly.
- HORIZON: It is a flight mode mixture between ANGLE and pure acrobatic.
- BARO: The barometer only is used, for the purpose of keeping a certain height.

- MAG: Heading lock mode. Can be activated in all flight stabilization methods.
- GPSHOME: Uses compass and GPS for the purpose of returning home to the starting point.
- GPSHOLD: Hold current position using GPS and baro (if available).
- HEADFREE: It only holds the orientation (yaw) of the multi and will always move in the same 2D direction for the same ROLL/PITCH stick movement.
- HEADADJ: Sets a new yaw origin for HEADFREE mode.

Output channel

In this window it is possible to check if the receiver is operational and if the range of each channel is correct. With receiver inputs responding one can check the channel polarity and range.

Output engines

Depending on the status of the card (not armed, armed, operating) it shows the output values of each channel of the motors.

Orientation

Graphical representation of the multicopter's attitude. In general, the main window of the Multiwii GUI, can be seen in Fig. 5.12

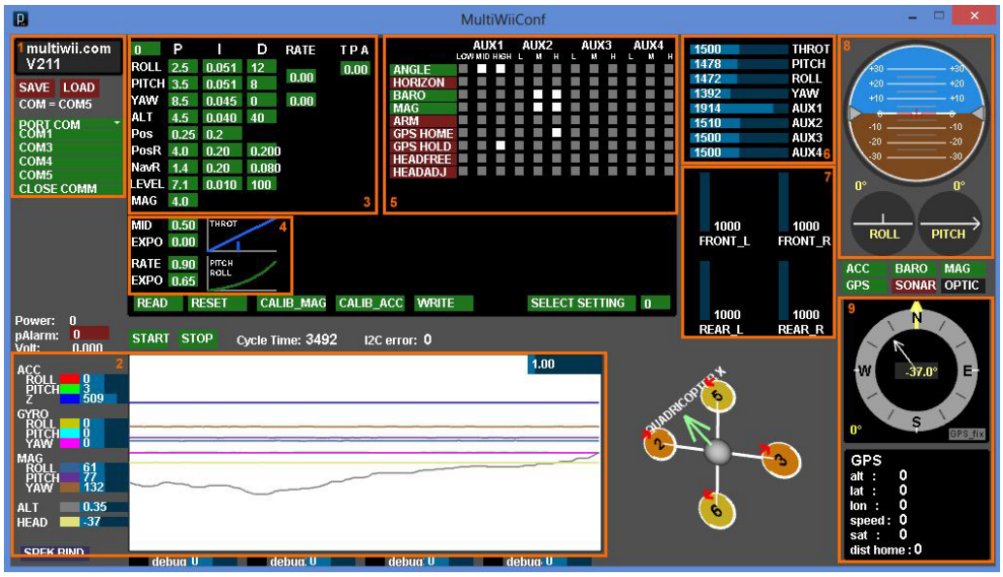


Figure 5.12: Multiwii GUI.

5.2.3.3 Sensors calibration

Once the card is connected and ready to use, it is necessary to perform a calibration of the sensors.

Accelerometer calibration

Place the board of the quadcopter flat on a table, plug the board into the USB connection of a computer and startup the MultiWii Config GUI, connect to the COM port and find the ACC sensor readings.

It should read close to 0 (within the range -10 to 10) for PITCH and ROLL, and close to either 256 or 512 in the Z axis. If it is outside this range then it is desirable to calibrate the ACC again, and find a more horizontally surface to do this.

Click the CALIB-ACC button above the graph section and wait a few seconds. The ACC sensor readings should now be close to the values in the above paragraph, so the ACC PITCH and ROLL near zero, and Z near 256 or 512. The Z value of 256 or 512 will depend on what sensor and firmware version are used, both values are valid.

Magnetometer calibration

Hit the CALIB-MAG button in the Multiwii GUI, this starts the calibration process. Since here, one have 30 seconds to rotate the board in all orientations. It does not matter which order of directions it is rotated.

Once the sensor is calibrated, the compass indicator in the MultiWii Config GUI should operate accurately as a normal compass. Try to rotate the copter through different directions and there should be no twitching, quick jumps in heading, or reversals.

5.2.3.4 Motor control

A common problem experienced with quadcopters is the time-variant thrust response due to a drop of the battery voltage. In other words for a same command received by the flight controller, the resulting thrust given by the motors will depend on the battery state of charge. This time-variant response of the quadcopter requires a high integrator gain in the position controller which results in adding too much phase. For the flexbot, and the nanoQx, the DC motors are controlled directly by the flight controller card through some transistors, but for hybride vehicle drone, as power is required for the motors, Electronic Speed Controllers (ESC) are used in order to provide the necessary amount of power and to handle the 3-phases of the motors. However, most of the ESCs do not achieve closed-loop control of the motor speed, and are then sensible to the voltage drop of the battery. BLHeli is an open source project

intended for replacing the official firmware of different ESCs. The main advantage is that it provides a sensorless closed-loop control mode of the motor speed. Therefore, the rotation speed should not be impacted by the battery's state of charge. Several adjustable parameters are available, however finding the good values for our setup is quite difficult without any objective measurement. For this reason, a test bench for the couple motor/ESC has been build in order to quantify the effect of the different tuning parameters.

5.2.3.5 Bench Setup

The only data we want to acquire is the speed of the motor in order to measure the response time and to check if the closed-loop control is able to reject a voltage drop. A hall sensor effect is used to measure the speed, for the considered model of the motor, the poles are directly visible by the sensor, if it would not be the case some magnets can be glued around the motor to trigger the sensor. An arduino *Uno* has been programmed to compute the frequency given by the hall sensor effect, this frequency is then send over USB to a PC periodically. The PC send several setpoints through USB to another arduino card which generates the PWM signal for the ESC, 1ms pulses translate to zero throttle, 2ms pulses are full throttle. Figure 5.14 shows the input profile sent to the ESC, the first part (from 0 to 10 seconds) consists in an ascending and descending ramp, the second part (from 10 to 20 seconds) tests several step response, the third part (from 20 to 22 seconds) consists in a high frequency reference, and during the last part, the voltage of the power supply is dropped by 1 Volt. A Labview real-time software have been built to send, receive and save the different data.

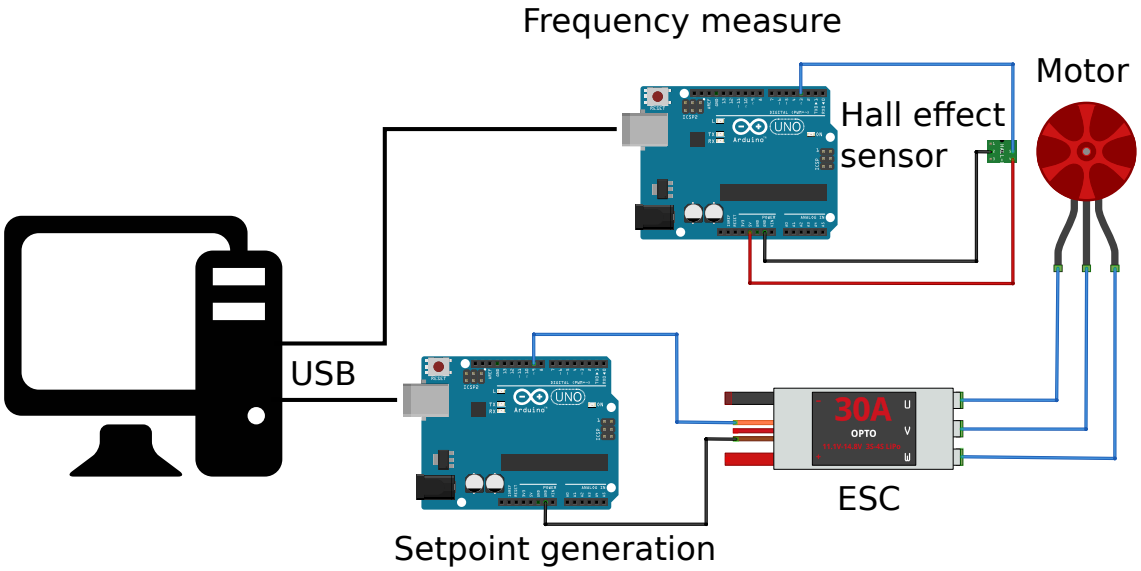


Figure 5.13: Test bench ESC + motor.

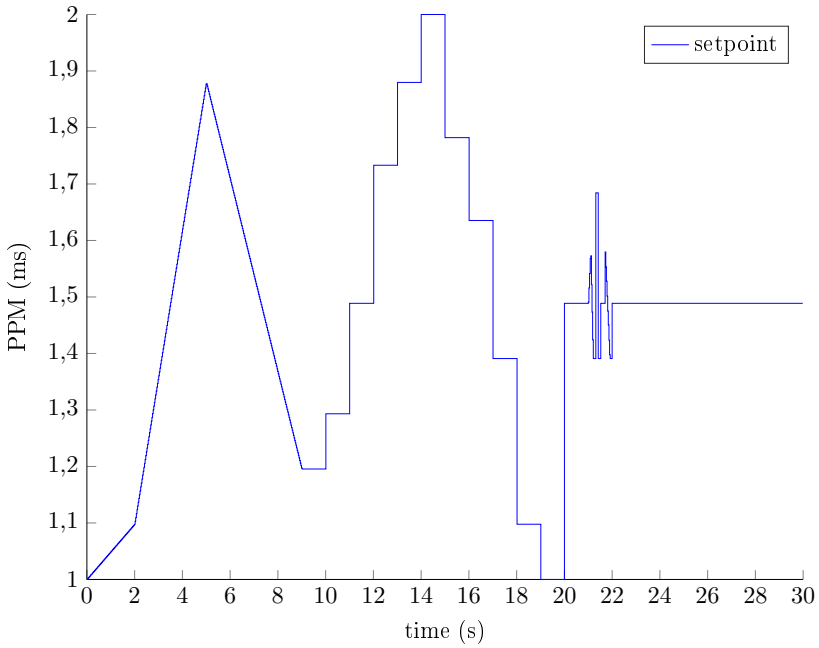


Figure 5.14: Profile input for the tuning of the motor control loop.

5.2.3.6 Results

18 configurations of parameters have been tested, Figure 5.15 shows the measured speed of the motor for 4 different tuning parameters. It can be seen that the response to the ramp input is the same for all the closed-loop parameters. To determine the best values for the proportional gain (Kp) and the integral gain (Ki), we will consider two important behavior: the rejection of a voltage drop and the step response.

- It can be seen clearly on Figure 5.16 that a voltage drop of 1 Volt (near 24 s) results in a speed drop of about 15 Hz for the open-loop control, whereas all the closed-loop responses are less impacted and tend to recover the speed drop. As the drop was manually applied to the power supply, it is not synchronized between all the experiments.
- Figure 5.17 shows the response to a step input, at $t = 11$s the PWM signal input of the ESC goes from 1.29 ms (29 % full speed) to 1.49 ms (49 % full speed). The open-loop response has not been plotted because it does not converge to the same value.

The parameters which have been found to give the best performances in terms of both disturbance rejection and step response are Kp = 3, and Ki = 3. A high Ki tends to provide faster disturbance rejection but it leads to an important overshoot of the step response if the Kp is not high enough.

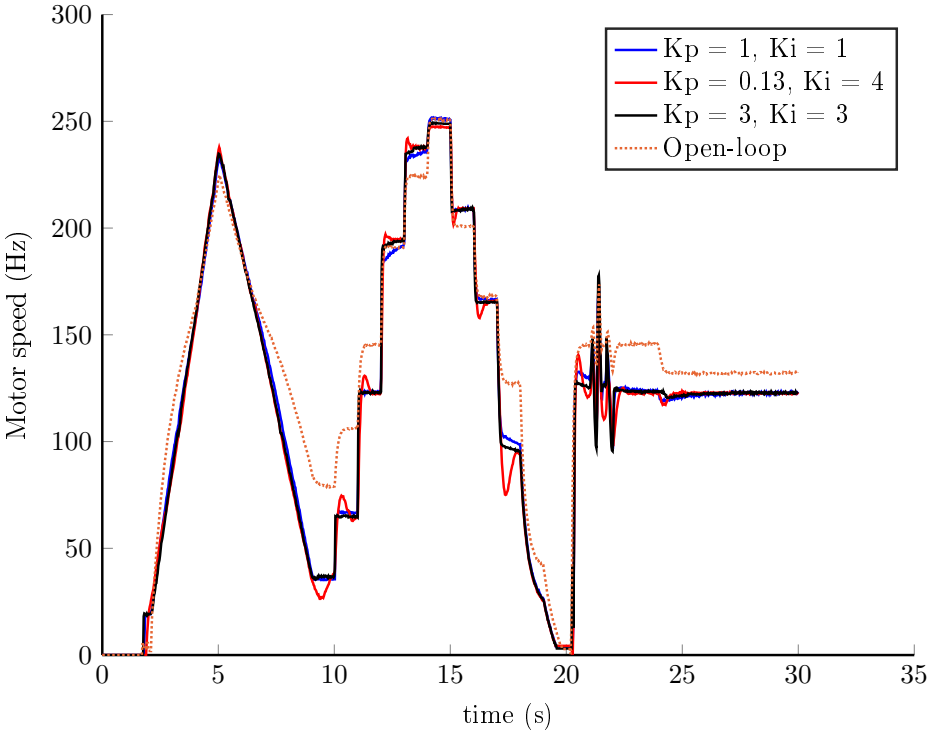


Figure 5.15: Measured speed for different tuning parameters.

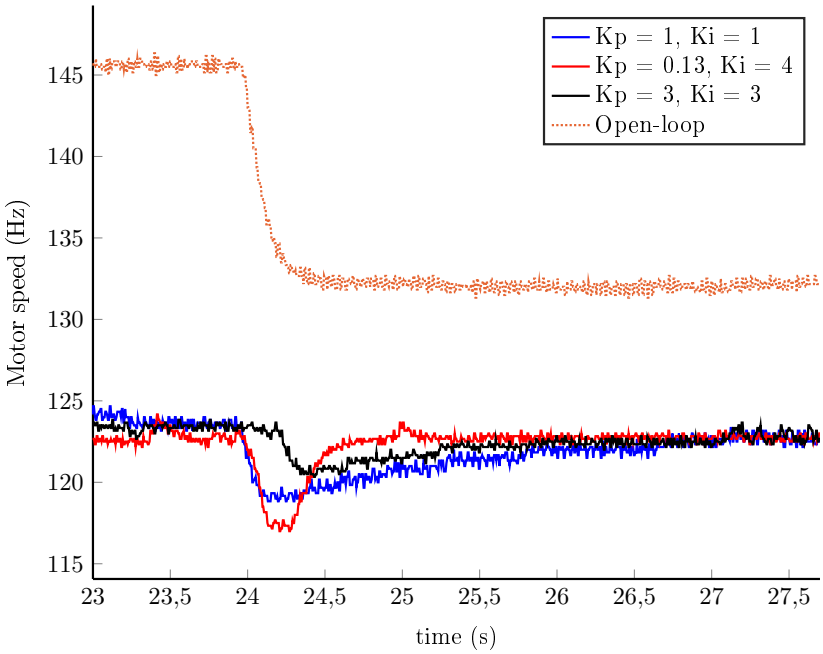


Figure 5.16: Disturbance rejection. 1 Volt drop is applied to the power supply, please note that the drop is not applied at the same time between the different experiments.

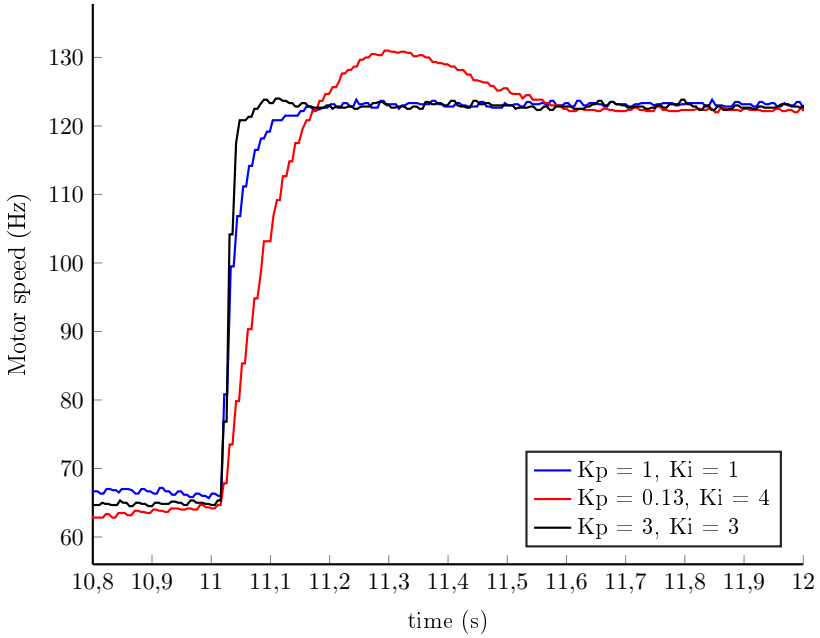


Figure 5.17: Step response. At $t = 11$s the PWM signal input of the ESC goes from 1.29 ms (29 % full speed) to 1.49 ms (49 % full speed)

5.3 Experiments on Ground Mode

The experiments in this section consist in the following a circular trajectory using the hybrid vehicle in ground mode. Two control algorithm were tested, the first one uses a nested saturation technique that allows to generate a force using the vehicle's propellers in the right direction in order to follow the desired direction. The take-off is limited by convenient values in the saturations. The other algorithm uses a non-linear technique that controls the yaw angle of the drone in order to point to the desired position. Then, according to the distance, the vehicle move faster if the distance to the target is great and slower if this distance get reduced. These control algorithms were described in previous chapters.

5.3.1 Nested Saturation Technique

The specifications of the vehicle used in this experiment are shown in Table 5.1. The experiment consist in following a circular trajectory of radius 0.7 m. at 0.4 rad/s. The front of the vehicle must point to follow the direction of the trajectory and thus the yaw angle will vary during the tracking. The parameters of the control algorithm are shown in Table 5.2. Figures 5.18 to 5.21 show the results of this test. The roll angle is about zero because the vehicle moves over a flat ground. The pitch angle lets the vehicle to move forward o backward depending on its sign and on the yaw angle. The thrust is lower than the minimum required for taking-off, that is, 3.4 N. The pitch angle is about a constant number because the required speed was constant as well. In Fig. 5.21 some variation are observed, this happened when the

Table 5.1: Hybrid Vehicle Parameters

Parameter	Value
mass	0.350 kg
payload	0.070 kg
length	0.32 m
height	0.30 m
helix diam	0.125 m
battery	1200 mAh 2S 7.4V 30C Li-Po
motor	Brushless 28000 kv 13 gr

vehicle tried to change its orientation in order to follow the trajectory. An error between the desired trajectory and one described by the vehicle is observed. In order to reduce this error, one could try to increase the parameters K_1 and M_1 but this led to deform the circular path made by the hybrid vehicle. We observe there exists a compromise during the tuning of the control parameters and we have chosen someones that make the vehicle to follow the circular path without so much deformation.

Table 5.2: Experimental parameters for saturation control in ground mode.

Parameter	Value	Parameter	Value
mass	0.350 kg	K_f	0.5
K_1	0.9455	M_1	0.31
K_2	0.54	M_2	0.27
K_p	1	K_q	1
ρ	5.5	K_{min}	0.5

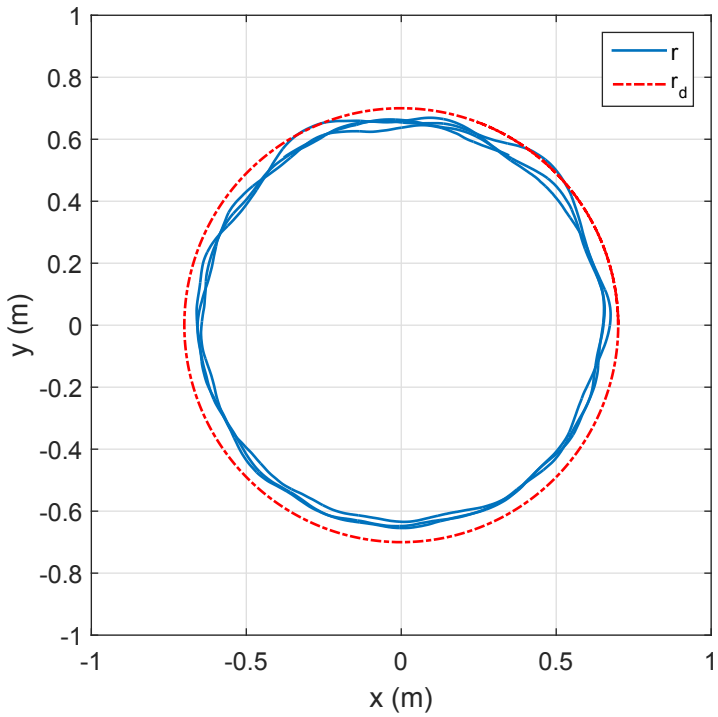


Figure 5.18: Trajectory performed by hybride vehicle in ground mode when following a circular path.

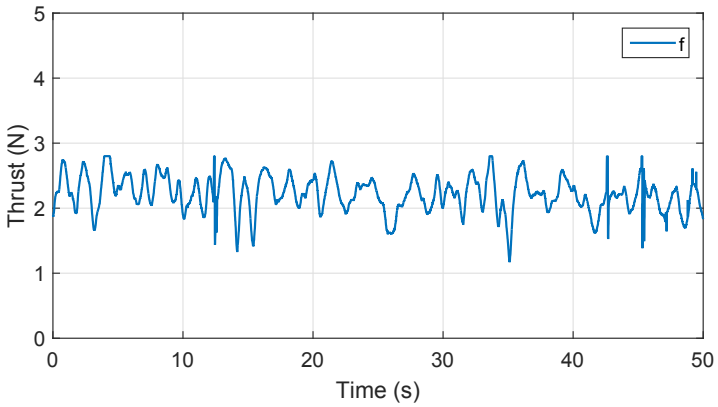


Figure 5.19: Thrust generated by helices when doing the circular path following.

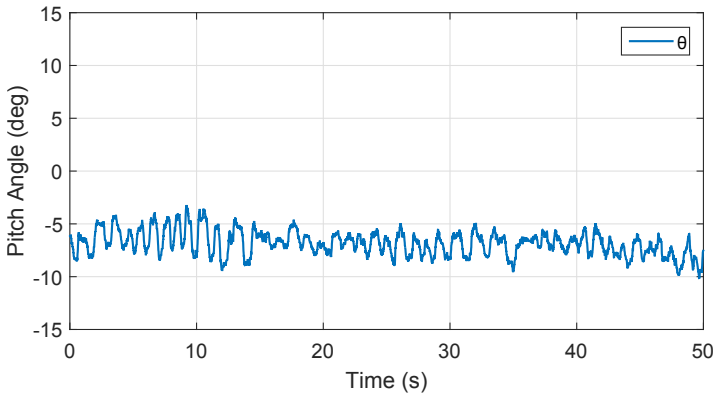


Figure 5.20: Pitch angle taken by the hybrid vehicle when following the circular path on ground

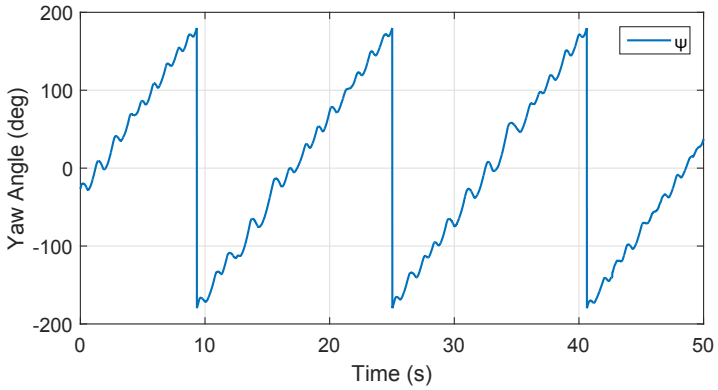


Figure 5.21: Yaw angle of the vehicle during the circular following in ground mode.

5.3.2 Nonlinear Algorithm

The test consist in the following of a circular path with radius 0.7 m. at a angular speed of 0.4 rad/s. The heading of the vehicle must be in the direction of the desired trajectory. The parameters of the vehicle are shown in Table 5.1 and the parameters of the control algorithm are presented in Table 5.3. The thrust of the vehicle is set to a constant value and the force needed for the displacement depends on the pitch angle. The orientation of the vehicle is controlled by the yaw angle and this must vary constantly when doing the following. The results of this test are shown in Figures 5.22 to 5.25. As it is observed the tracking error is almost zero and the form of the trajectory made by the vehicle is like a perfect circle. The heading of the vehicle is changing in order to adopt the right orientation during the following. The constant angular speed and the constant thrust of the vehicle drive to a pitch angle of about -5. In summary, this technique displays good performance when tracking a path over the ground.

Table 5.3: Experimental parameters for non-linear control in ground mode.

Parameter	Value	Parameter	Value
mass	0.350 kg	K_p	1
K_d	0.4	f	2.8 N

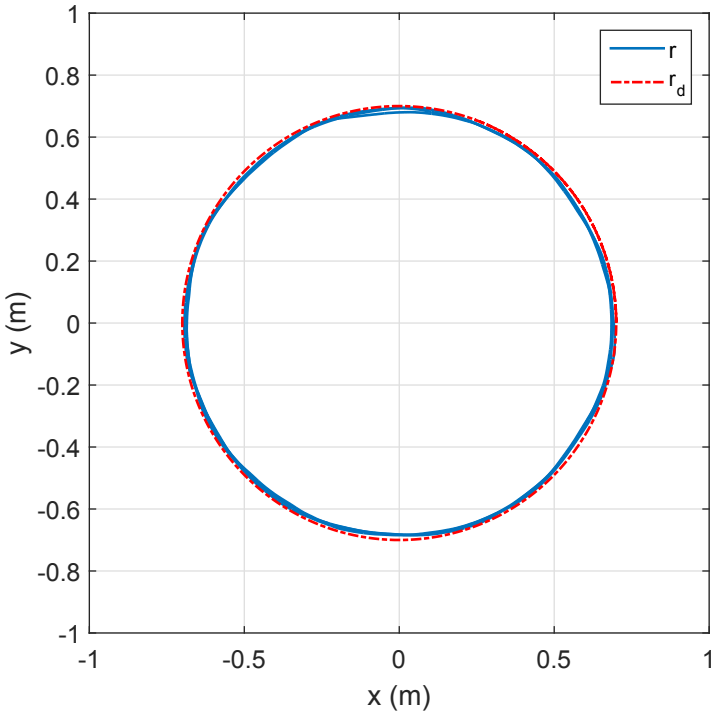


Figure 5.22: xy trajectory of hybride vehicle using a non-linear control law.

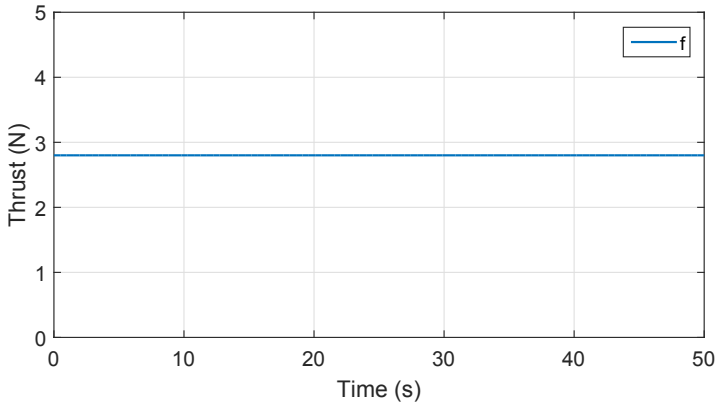


Figure 5.23: Constant thrust generated by propellers in the non-linear control law.

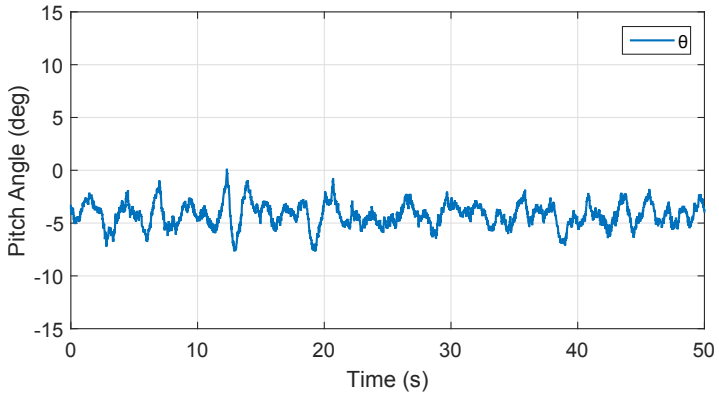


Figure 5.24: Pitch angle regulates the speed of the vehicle.

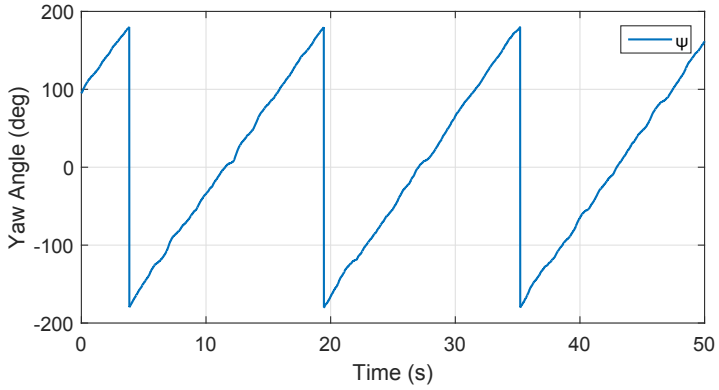


Figure 5.25: Yaw angle representing the vehicle's orientation during the following of a circular path using a non-linear control algorithm.

5.4 Experiments as a Quadrotor

There are several control laws that could be applied to the vehicle behaving as a quadrotor. One of the techniques tested in this section was the backstepping technique with adaptive

and integral properties. The adaptive control allows to estimate a slow perturbation and to compensate it by including the estimation. The integral properties helps to overcome the undesired effects of dynamics not modeled or neglected. Another technique considered is the nested saturation because that could be used to produce smooth movements. Finally, a quaternion based approach is tested and validated, this approach could be useful when the vehicle need to move in complex trajectories avoiding the limitations of the Euler angles.

5.4.1 Backstepping with adaptive property

The drone used in this experiment is the Blade NanoQX showed in Fig. 5.4 and its specifications are described in Table 5.4. In this experiment, a load of 4.7 grams is added to the drone, which is about 25 % of vehicle weight. The first test does not include the adaptive part and second one does include it. In general, the drone without the adaptive property does not well manage the presence of a disturbance. The quadrotor stayed at a low altitude during the test and was unable to achieve the reference position. When the control algorithm includes the disturbance estimation, the vehicle has a better performance. Fig. 5.26 shows that error tends to zero and consequently the thrust required is greater as shown in Fig. 5.27. The load estimation is shown in Fig. 5.28 and particularly, note $\hat{k}_{uz}$ agrees with the weight added. This estimation also helps to compensate the degradation of the control performance due to the battery discharge.

Table 5.4: NanoQX Parameters

Parameter	Value
mass	20 gr
payload	6 gr
Transmitter	MLP4DSM
Length	0.140 m
Height	0.030 m
helix diameter	0.05 m
Battery	150mAh 1S 3.7V 25C Li-Po
Motor	6 mm Brushed

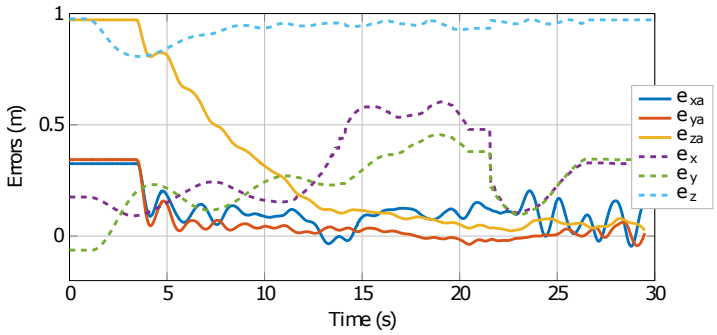


Figure 5.26: Position error with adaptive algorithm (subscript a) and without the adaptive property when a load of 25 % of its weight is added.

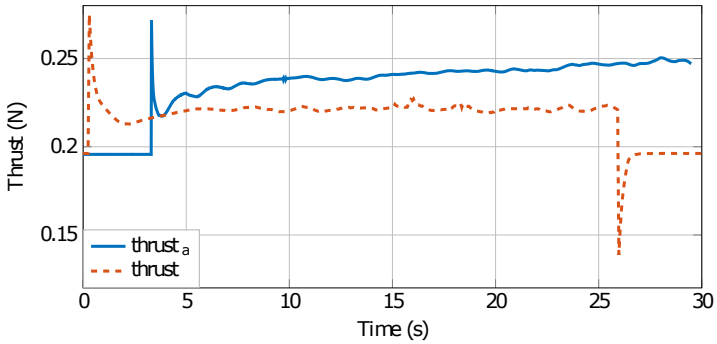


Figure 5.27: Thrust comparison with and out without the adaptive property (subscript a means adaptive).

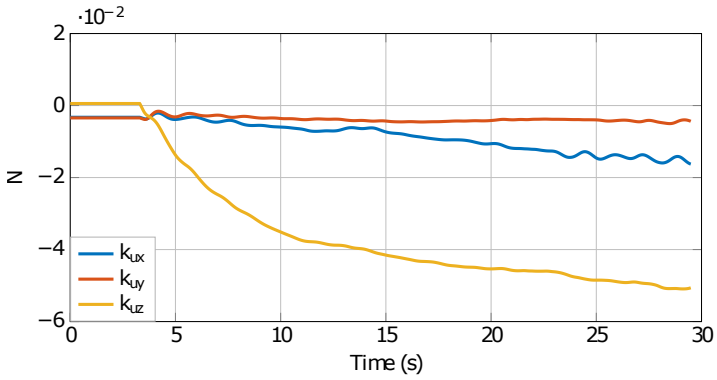


Figure 5.28: Estimation of the load and of the not modeled dynamics.

5.4.2 Backstepping with integral property

The vehicle used during the test of this control law is a quadrotor, that is the hybrid vehicle without the wheels and hence its specifications are similar with those presented in Table 5.1. The weight of the vehicle is 280 gr. and the payload about 140 gr. The test consists in doing a circular path at a height of 1 m. with the vehicle front pointing in the movement direction and at a speed of 0.6 rad/s. The position control law has an integral property that helps to

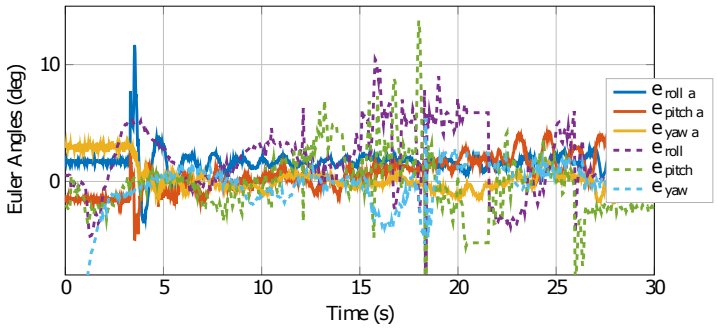


Figure 5.29: Euler angles evolution with and without adaptive compensation. With adaptive compensation, the variation is less important.

improve the following of the trajectory and the attitude part has adaptive properties. The parameters' values for the position control algorithm used during this test are shown in Table 5.5.

Table 5.5: Experimental parameters for position control of a quadrotor based on an integral backstepping technique.

Parameter	Value	Parameter	Value
mass	0.280 kg	$\mathbb{K}_{Ir}$	diag(0.01, 0.01, 0.01)
$\mathbb{K}_r$	diag(1.1, 1.1, 1.1)	$\mathbb{K}_v$	diag(1.8, 1.8, 0.45)

The results are shown in Figures 5.30 to 5.33. These results validate the good performance of the backstepping with integral property. The quadcopter rise to 1 m. and in order to move along the circular path it changes its yaw, pitch and roll angles. The variation of pitch and roll angles are not very big because the required speed is not very big.

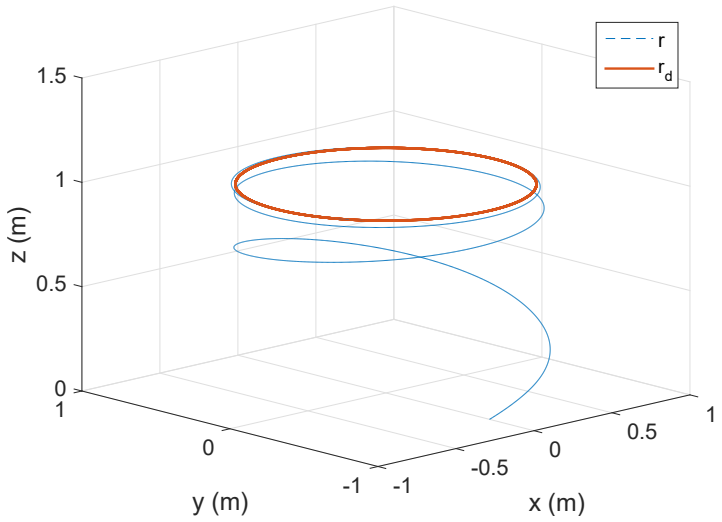


Figure 5.30: Trajectory described by the quadrotor using an adaptive integral backstepping.

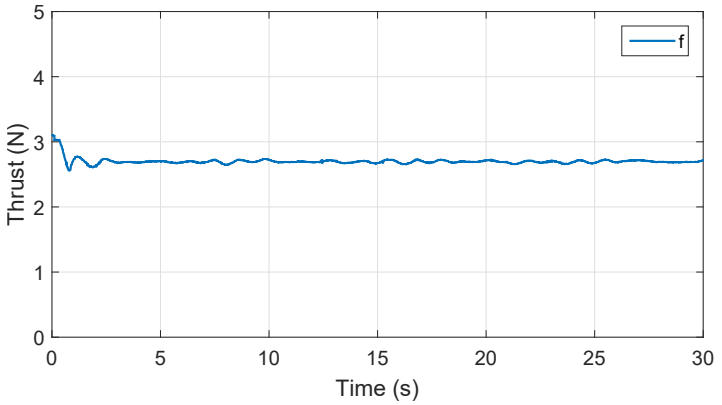


Figure 5.31: Thrust generated by helices during the flight.

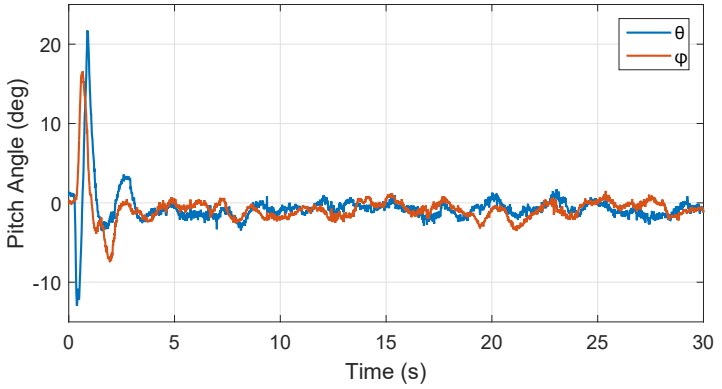


Figure 5.32: Evolution of the pitch and rolll angle when doing the circular path following.

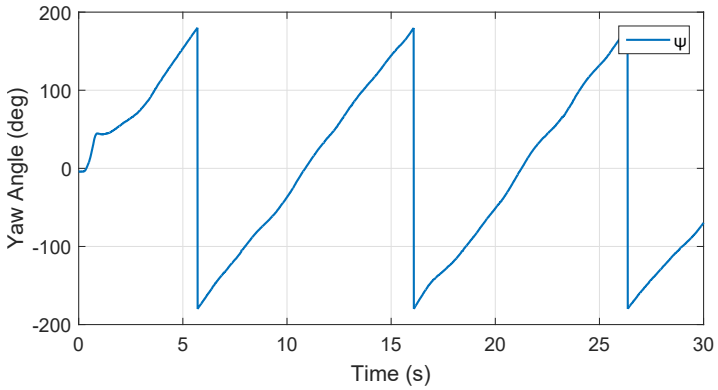


Figure 5.33: Yaw angle of the quadorotor changes according to the direction of the movement.

5.4.3 Nested Saturation

In this test, the nested saturation technique is used for a position control. The position reference is $(x, y, z)_{ref} = (0, 0, 0.5)$ m. This test uses the hybrid vehicle as a quadrotor and the parameters' values for the position control are shown in Table 5.6.

Table 5.6: Experimental parameters for position control of a quadcopter based on a nested saturation algorithm.

Parameter	Value	Parameter	Value
mass	0.350 kg	$\mathbb{K}_1$	diag(1.5, 1.5, 1.5)
$\mathbb{M}_1$	diag(1, 1, 1)	$\mathbb{K}_p$	diag(1, 1, 1)
$\mathbb{K}_2$	diag(1, 1, 1)	$\mathbb{M}_2$	diag(0.5, 0.5, 0.5)
$\mathbb{K}_q$	diag(1, 1, 1)		

Figures 5.34 to 5.36 show the vehicle converges to the desired position. A small following error appears during this test because this saturation technique does not include any integral or adaptive property to reduce the effect of the dynamics not considered in real world. This error is also reflected in the Euler angles, the algorithm tries to compensate this error and it needs to adopt a certain orientation in order to stay near to the desired position.

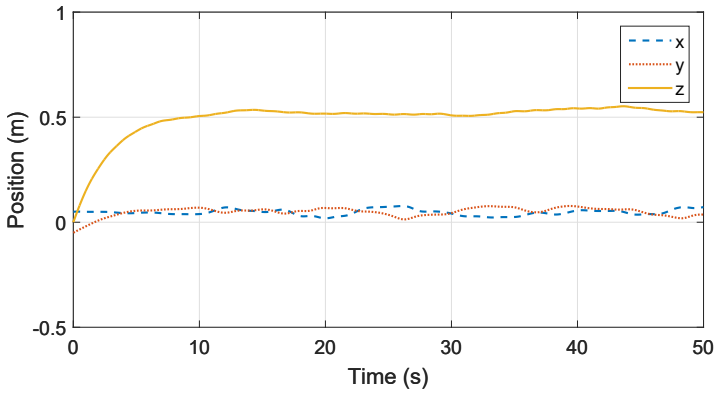


Figure 5.34: Position of the quadrotor using an algorithm based on a nested saturation.

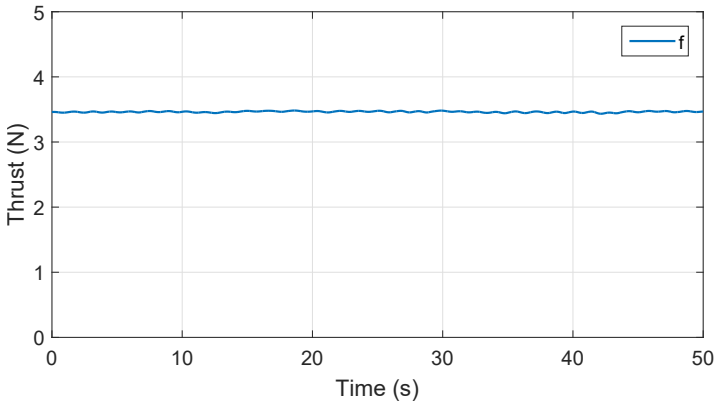


Figure 5.35: Thrust employed by the saturation control.

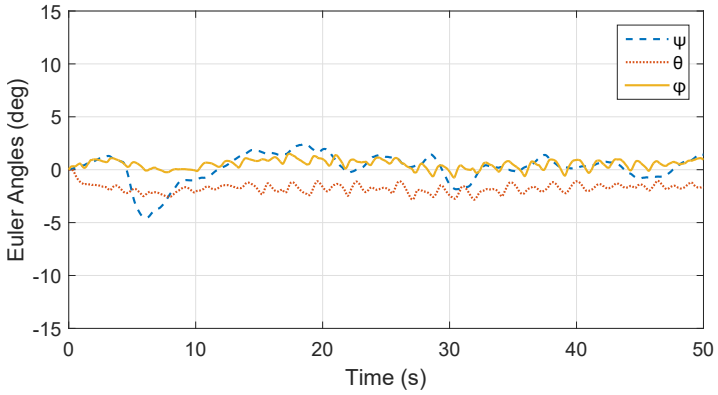


Figure 5.36: Orientation adopted by quadrotor in order to reach the desired position using a saturation-based algorithm.

5.4.4 Intermediary Quaternion

This test uses the hybrid vehicle but without wheels. The experiment consist in following a circular trajectory placed at $(x, y, z) = (0, 0, 0.5)$ m. and radius 0.7 m. at a speed of 1.6

rad/s. This time the required yaw is in such a way the vehicle front is pointing to the circle center. The parameters' values are fixed to those shown in Table 5.7. The orientation control algorithm was programmed in the flight card of the prototype.

Table 5.7: Experimental parameters for position control of a quadrotor based on intermediary quaternions.

Parameter	Value	Parameter	Value
mass	0.260 kg	$\mathbb{K}_{Ir}$	diag(0.09, 0.09, 0)
$\mathbb{K}_r$	diag(0.9, 0.9, 0.75)	$\mathbb{K}_v$	diag(0.9, 0.9, 0.5)

The results ares shown in Figures 5.37 to 5.40. The trajectory converge to the path reference and turning at the desired speed. The intermediary quaternion error converges to the $\boldsymbol{q_e} = 1 + \mathbf{0}$ which means the orientation converge to the required attitude by the control algorithm in order to describe the circular path. The pitch, roll and yaw angles shown in the figures gives a physical sense of the movement. This technique does not have the singularity issue of Euler approach and also is easy to compute from the rotation matrix. Additionally, the unwinding phenomenon is not present.

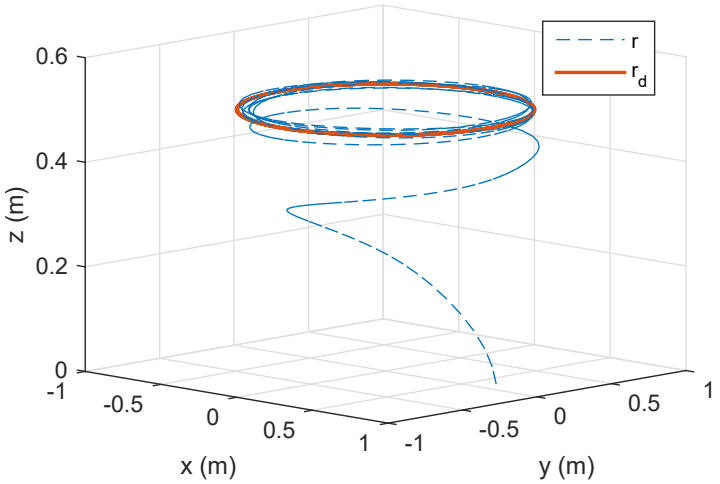


Figure 5.37: Trajectory described by the quadrotor using a intermediary quaternion control. Quadrotor trajectory follows the circular path reference.

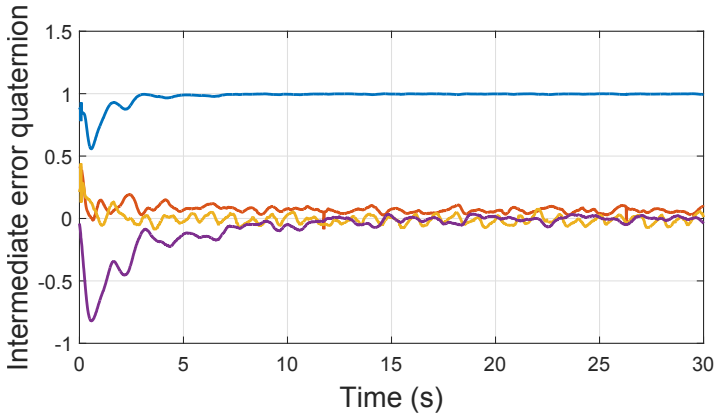


Figure 5.38: Quaternion error tends to the identity, that is, the orientation converges to the required attitude asked by the position control.

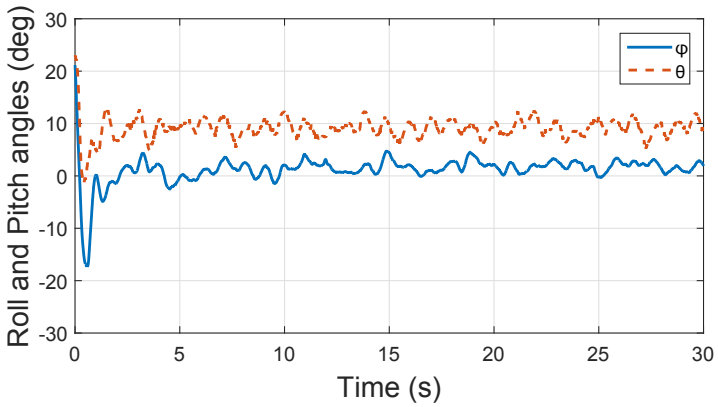


Figure 5.39: Behavior of the pitch and roll angles. Note the roll angle takes a constant value when doing the circular path and the pitch angle is slightly above zero in order to counteract the centrifugal force during the turns.

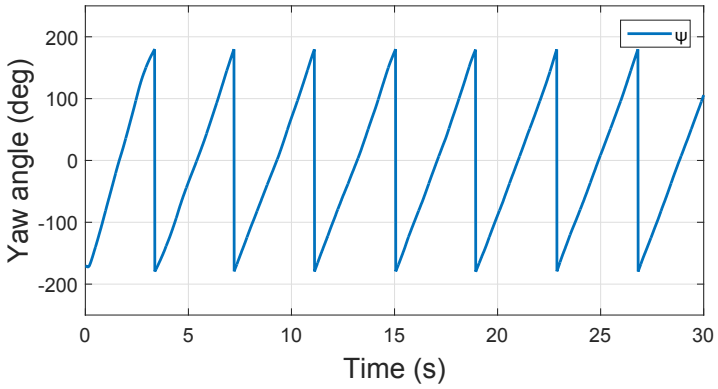


Figure 5.40: Yaw angle indicates the vehicle front is pointing to the circle center during the flight.

5.5 Experiments as a Hybrid Vehicle

In this test the hybrid vehicle uses only the non linear control with proportional derivatives properties.When the vehicle passes from flight operation mode to ground mode or vice-versa the algorithm changes only the values of its parameters. The first stage will be the tunning of the parameters for each operation mode and the next stage the switching between the two values sets. The trajectory to follow by the vehicle is composed by four parts. The first one is a sinusoidal curve beginning at $(x_0, y_0, z_0) = (1, 0, 0.1)$ m and ending at $(x_1, y_1, z_1) = (1, 0.3, 0.7)$, then the second curve is a circle that begins at the end of the first curve and ends at $(x_2, y_2, z_2) = (-1, 0.3, 0.7)$. The third curve is a sinusoid beginning at the last point of second curve and finishing at $(x_3, y_3, z_3) = (-1, 0, 0.1)$ and the last curve is a circle that begins at this point and ends at the beginning of the fist curve.

The parameters of this vehicle are those shown in Table 5.1. The only remark is the height of the vehicle's geometric center is about 15 cm and hence the vehicle cannot attain the desired height $z_d = 0.1$ m because the ground prevents it. The two sets of parameter values for position control used in this experiment are:

$$
\begin{array}{ll}
k_{va} = 0.75 & \\
k_{pa} = 0.1 & k_{pg} = 0.1 \\
k_{da} = 0.1 & k_{dg} = 5 \\
k_{pz} = 0.7 & f_g = 2.8\ Nm \\
k_{dz} = 0.4 & m = 0.350\ kg \\
k_{iz} = 0.01 &
\end{array}
$$

where the subscript a means aerial and g stands for ground. The results are shown in Figures 5.41 to 5.44. The switching from aerial to ground mode is made when the desired height is lower than 10.2 cm. In ground mode the thrust is constant and limited in order prevent the take-off. The vehicle front follows the direction of the movement and is almost constant during the take-off and landing.

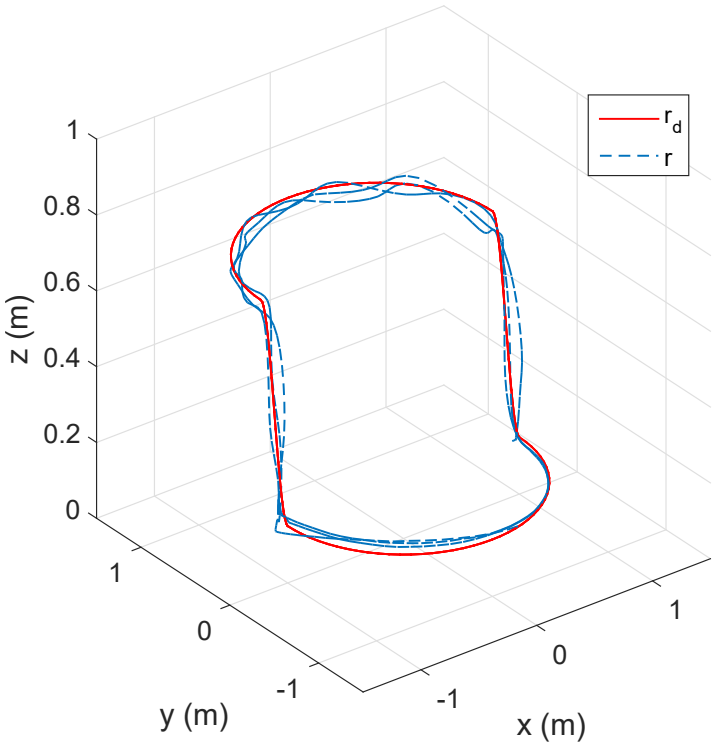


Figure 5.41: Trajectory described by hybrid vehicle when doing the following of a hybrid terrestrial and aerial path.

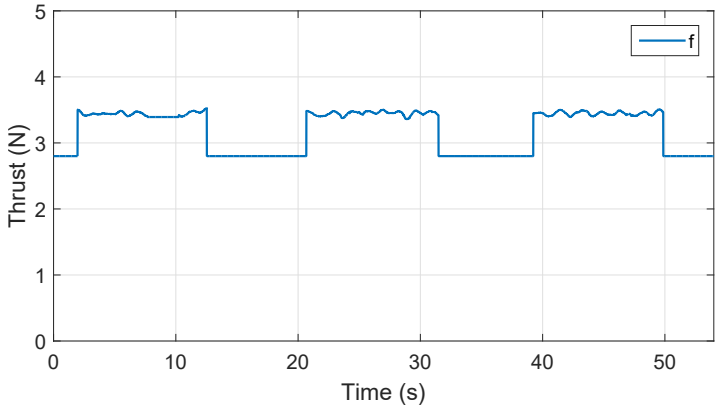


Figure 5.42: Thrust of the vehicle changes according to the operation mode. In ground mode the thrust reduces to 2.8 N in order to avoid the take-off.

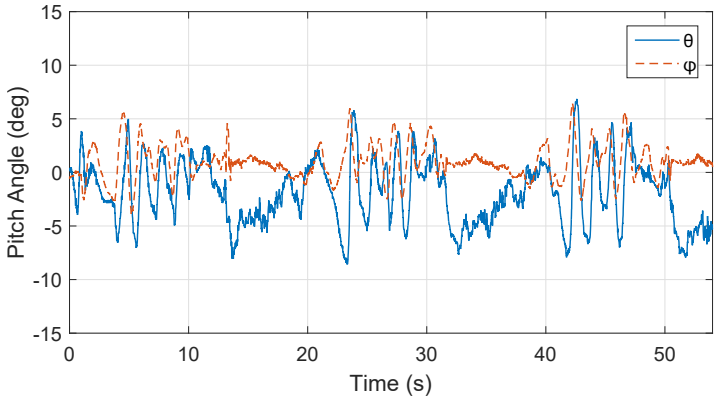


Figure 5.43: Pitch and roll angles described during the following of the hybrid path. Over ground, the roll angle is almost zero and the pitch angle changes in order to make move the vehicle.

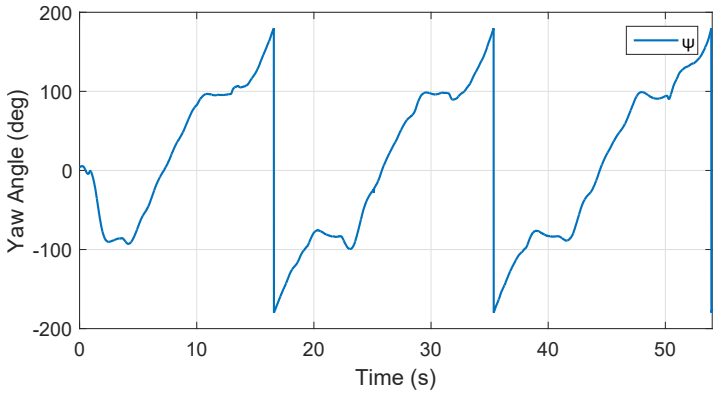


Figure 5.44: The yaw angle changes according to the stage during the following. During the take-off and landing the vehicle exhibits an almost constant values.

5.5.1 Ground detection with Disturbance Observer

In this experiment, the vehicle tracks a hybrid terrestrial an aerial path but the switching from aerial mode to ground mode is made by the detection of the ground contact. When the vehicle touches the ground there appears a reaction force acting on the vehicle wheels. Therefore, the idea is to know the magnitude of this force in order to know the contact with the ground. The parameters values for each mode and the characteristics of the vehicle are the same as in the previous section 5.5. The results are shown in Figures 5.45 to 5.49. When the observer gives a value of 0.165 N. the vehicle changes to ground mode. The other graphics described a similar behavior as obtained previously without the observer. Therefore, this experiment validates the use of the disturbance observer to detect the ground contact and to switch from aerial to ground mode.

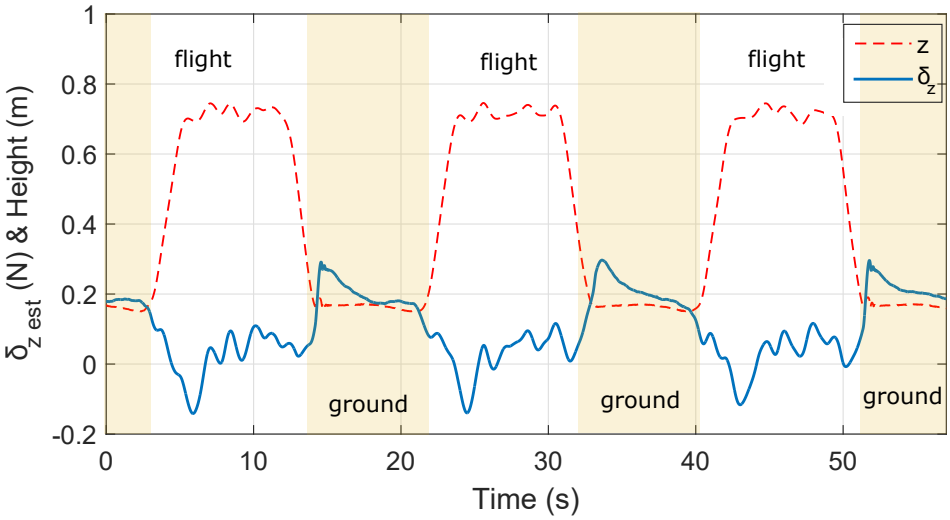


Figure 5.45: Disturbance observer estimating the force acting on the vehicle during the tracking of the hybrid path. When the vehicle is in contact with ground, its estimation increases over 0.165 N. and this lets to determine the moment of ground contact.

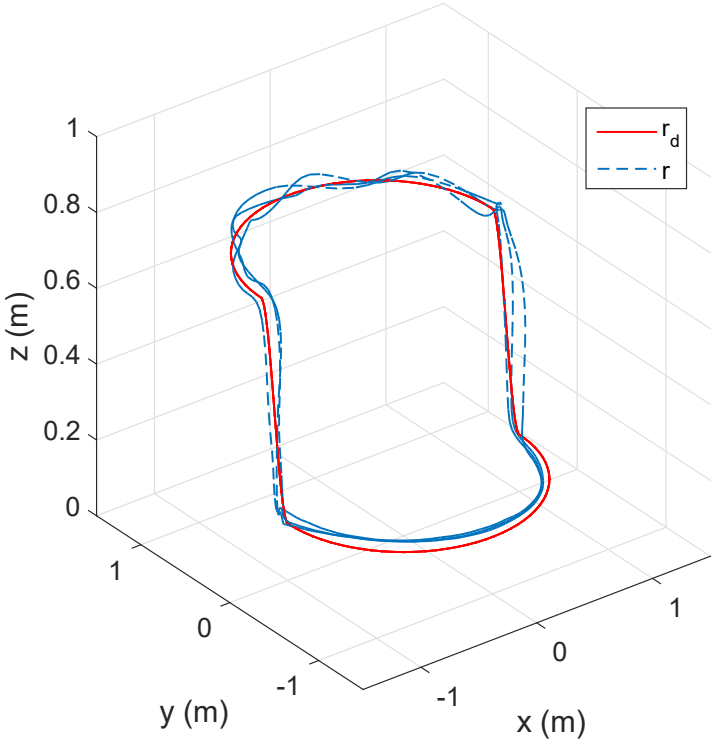


Figure 5.46: Trajectory followed by hybrid vehicle when doing the tracking of a hybrid terrestrial and aerial path.

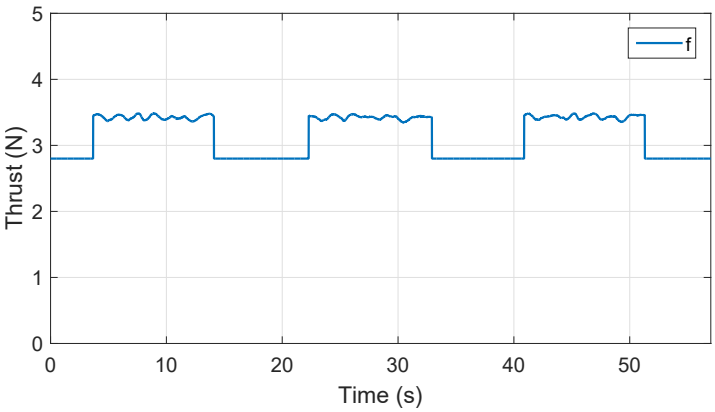


Figure 5.47: Vehicle thrust changes during the following.

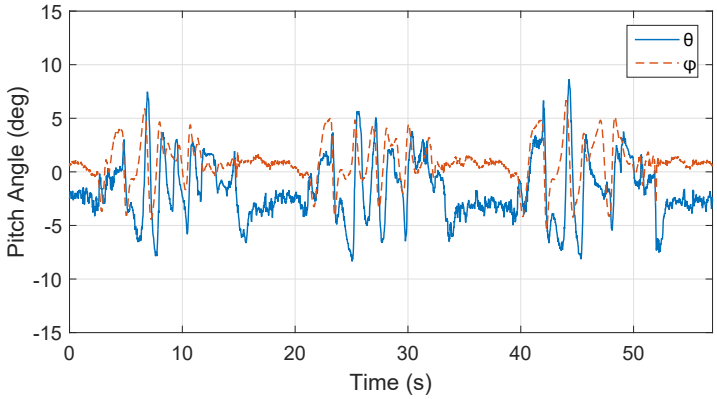


Figure 5.48: Evolution of the pitch and roll angles during the following of the hybrid path.

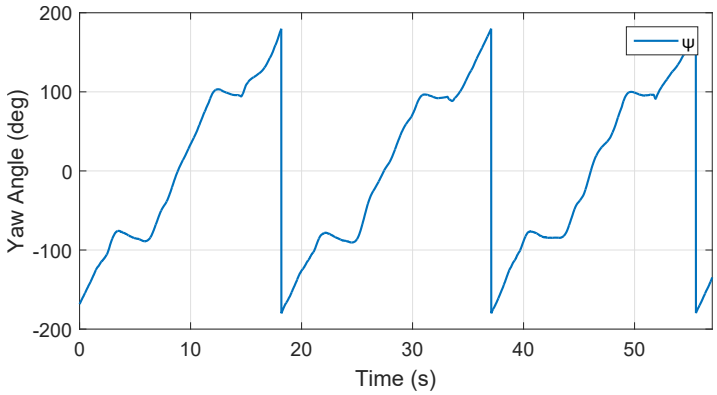


Figure 5.49: The yaw angle changes according to its position during the following.

5.6 Summary

This chapter described the MOCA room at GIPSA-Lab and the equipment used for the experimental tests. Also a survey of the prototypes used during the experiments was presented. Beginning with a commercial drone and finishing with a 3D printed drone with an open flight controller adapted to our needs. Several issues were faced during the tests, mainly the short operation time of early prototypes and the variation of the rotation speed of the motors due to voltage drops. Then, it was described the experimental tests in ground and aerial mode. These tests validated the results previously confirmed in numerical simulations. Finally, the non linear control with proportional and derivative properties is used in the tracking of an aerial and terrestrial trajectory. Firstly, the adequate values for the parameters in both modes were found and next they were switched when the vehicle passed from air to ground and vice-versa. Lastly, the disturbance observer was used in order to identify the moment when the vehicle touched the ground. This knowledge let to make the switching automatically from air to ground operation without additional sensors. Some videos [61] were included so that the reader can watch these algorithms tested in MOCA room.

Conclusion and Future Works

This work was born from the idea to conceive a vehicle that can fly but also that can act as a little car. Reviewing the present works about vehicles with these characteristics, they uses wings or small supports to make terrestrial displacements, others use a sort of cage in order to rolling over the ground. The first ones make slow displacement but the other ones move faster. Another characteristic is the reuse of the same structure for operating in two modes for being more energy efficient. These observations led us to conceive a particular vehicle that reuses the same structure, be energy efficient, reduces contact point with ground and, in one word, improves the present prototypes. The **Chapter 1** is focused in a revision of theses vehicles and introduced the proposed one. The interest on a hybrid vehicle is because it can accomplish tasks more complex, it can adopt the operation mode more convenient depending on the environment. Also, the range of potential use of these vehicles increases, for example, they can serve to carry out explorations, inspections of irregular places, damaged building by earthquakes, rolling over inclined surfaces, avoiding obstacles or doing mixed trajectories among others. Once the vehicle was conceived, the next step was to understand its physical behavior. The remaining sections of the first chapter exposed some fundamental mathematical background that helps to understand the dynamics of a rigid body in rotation.

The **Chapter 2** developed the model of the hybrid vehicle. First, the ground operation mode is analyzed and a model for this mode is deduced. Second, the aerial mode is treated and the corresponding model is obtaining by applying the Euler Newton approach. Finally, making a revision of the models for each operation mode, a model for the hybrid vehicle is proposed. This model gathers the characteristics of both models and, under specific assumptions, it is possible to reduce to anyone mode. Also the conditions for each model holds are presented. Once, the physical dynamics is described by a mathematical model, the challenge is to design one o more algorithm in order to control the movement of the hybrid vehicle.

Chapter 3 considered some techniques for controlling the vehicle. In ground mode, two strategies are considered. One is based on a nested saturation control because the saturations allow to limit the thrust of the vehicle and by consequent it prevents the take-off of the terrestrial vehicle. The other one was born from a natural observation, that is, the vehicle must be aligned with the direction to the target position and to move forward. The behavior of the displacement depends on the distance and direction to the target. In further sections, the control algorithms for the aerial mode are developed. Expressions for backstepping with integral and adaptive properties are computed. Also the nested saturation technique is considered because it could be useful for generating smooth movements. Furthermore, in order to develop an attitude control algorithm that could serve to make more complex trajectories, a control law based on intermediary quaternions is computed. Finally, a high disturbance observer is adapted to our system in order to identify the external forces acting on the vehicle or to identify the dynamics resulting from assumptions in the model. Also the use of this observer could help to improve performance of some control algorithm.

The next stage was to validate the control algorithm considered in the previous chapter. **Chapter 4** is dedicated to corroborate these algorithms. The values introduced in the simulations were chosen as near as possible the the real ones in order to get an insight before the experimental tests. Circular trajectories were proposed for the two operation modes with some differences in every simulation. External forces and noisy measurement were considered as well, in order to emulate the real world and to prove the pertinence of each control law. The results obtained in this chapter validated the proposed algorithms and provided an preview to the real behavior.

The **Chapter 5** presented the experimental tests performed in the MOCA room. This room is equipped with VICON cameras that use infrared light to identify markers that reflect this specific light. This system gets the attitude and position of a body with markers at 500 Hz. From the beginning of these experiments, these were a rich source of experience and knowledge that help us in the next experiments. There were crashes of prototypes and humans errors were done but all this contributed to deepen in the world of real behavior of engines.

The first prototype was a NanoQX drone, very small and with a smooth fly. The first limitations appeared, the drone was not adapted to roll over ground, its payload was very limited. The next prototype let us to modify the attitude algorithm according to our requirements. The limitation was the payload and the little operation time. In the search of a customized prototype, the Crius AIO pro was used as flight controller and a 3D printed frame was used for the vehicle. Next stage was to validate some control techniques and to enhance the structure of the vehicle.

The control laws were validated in simulations and also in experimental tests. It was noticed that some algorithms works better than others depending on the operation mode. A control law that could be used in both modes was chosen. The algorithms based on nested saturations and the nonlinear control with proportional and derivative properties fulfill this requirement. The non linear control was tested with a hybrid aerial and terrestrial trajectory. Firstly, the switching between the parameters values for each mode were made according to the desired trajectory the vehicle needed to cover. Secondly, the implementation of an observer to identify the moment of ground contact allowed to make the switching between both different sets of values.

It was observed the performance of the vehicle could be improved if the tunning of parameters were been made at different heights. The behavior of the vehicle when it is flying near to the ground is different to when it is flying far. The ground effect was not considered.

A next stage will be the identification of the disposition of surfaces which are in contact with the vehicle. This identification could be carry out by means of the employed high order observer. This could lead to adapt the parameters values in order to move over inclined surfaces, to mount walls or to move attached to the ceiling.

Another future work is the evaluation of the nested saturation control in hybrid trajectories with an intermediary quaternion based control for its attitude and allowing more complex maneuvers.

Finally, as result of the development of this work, some publications were produced [22, 62–64], namely,

- Conference papers
  - Integral Backstepping Control for Trajectory Tracking of a Hybrid Vehicle, ICUAS 2015
  - Position Control of a Quadrotor under External Constant Disturbance. REDUAS 2015
  - An Intermediary Quaternion-based Control for Trajectory Following Using a Quadrotor. IROS 2017
- Book participation
  - Nonlinear algorithms with integral action Indoor Navigation Strategies for Aerial Autonomous Systems. Chapter 6, Elsevier 2016.

Also, some contributions were made in related works [18, 65],

- Conference papers
  - Nonlinear Control of a Nano-Hexacopter Carrying a Manipulator Arm. IROS 2016.
  - Velocity control of mini-UAV using a helmet system. REDUAS 2015.

Bibliography

[1] L. Daler et al. "A flying robot with adaptive morphology for multi-modal locomotion". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 1361–1366 (cit. on p. 2).

[2] S. Thorel and B.d'Andréa-Novel. "Hybrid Terrestrial and Aerial Quadrotor Control". In: *IFAC Proceedings Volumes*. Vol. 47. 3. 2014, pp. 9834–9839 (cit. on p. 2).

[3] C. J. Dudley, A. C. Woods, and K. K. Leang. "A micro spherical rolling and flying robot". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 5863–5869 (cit. on p. 3).

[4] A. Kalantari and M. Spenko. "Design and experimental validation of HyTAQ, a Hybrid Terrestrial and Aerial Quadrotor". In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 4445–4450 (cit. on p. 3).

[5] R. J. Bachmann, R. Vaidyanathan, and R. D. Quinn. "Drive train design enabling locomotion transition of a small hybrid air-land vehicle". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 5647–5652 (cit. on p. 4).

[6] K Peterson et al. "A wing-assisted running robot and implications for avian flight evolution". In: *Bioinspiration and Biomimetics* 6.4 (2011), p. 046008 (cit. on p. 4).

[7] E. Canigur and M. Ozkan. "Model reference adaptive control of a nonholonomic wheeled mobile robot for trajectory tracking". In: *2012 International Symposium on Innovations in Intelligent Systems and Applications*. 2012, pp. 1–5 (cit. on p. 15).

[8] W. Leroquais and B. d'Andrea Novel. "Modeling and control of wheeled mobile robots not satisfying ideal velocity constraints: the unicycle case". In: *Proceedings of 35th IEEE Conference on Decision and Control*. Vol. 2. 1996, 1437–1442 vol.2 (cit. on p. 15).

[9] B. d'Andrea Novel, G. Bastin, and G. Campion. "Modelling and control of nonholonomic wheeled mobile robots". In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. 1991, 1130–1135 vol.2 (cit. on p. 15).

[10] Yu Tian, N. Sidek, and N. Sarkar. "Modeling and control of a nonholonomic Wheeled Mobile Robot with wheel slip dynamics". In: *2009 IEEE Symposium on Computational Intelligence in Control and Automation*. 2009, pp. 7–14 (cit. on p. 15).

[11] J. R. Page and P. E. I. Pounds. "The Quadroller: Modeling of a UAV/UGV hybrid quadrotor". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, pp. 4834–4841 (cit. on p. 15).

[12] A. Azzam and Xinhua Wang. "Quad rotor arial robot dynamic modeling and configuration stabilization". In: *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*. Vol. 1. 2010, pp. 438–444 (cit. on p. 19).

[13] Y. Naidoo, R. Stopforth, and G. Bright. "Development of an UAV for search amp; rescue applications". In: *AFRICON, 2011*. 2011, pp. 1–6 (cit. on p. 19).

[14] R. Mahony, V. Kumar, and P. Corke. "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor". In: *IEEE Robotics Automation Magazine* 19.3 (2012), pp. 20–32 (cit. on p. 19).

[15] P. Castillo, R. Lozano and A. E. Dzul. "Modelling and Control of Mini-Flying Machines". In: London: Springer-Verlag, 2005. Chap. 2,3 (cit. on p. 19).

[16] Janet D. Finch, Louis N. Hand. "Analytical Mechanics". In: Cambridge: Cambridge University Press, 1998. Chap. 1 (cit. on p. 19).

[17] E. Fresk and G. Nikolakopoulos. "Full quaternion based attitude control for a quadrotor". In: *Control Conference (ECC), 2013 European*. 2013, pp. 3864–3869 (cit. on p. 19).

[18] J. U. Alvarez-Muñoz et al. "Nonlinear control of a nano-hexacopter carrying a manipulator arm". In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. 2015, pp. 4016–4021 (cit. on pp. 27, 121).

[19] H. Liu and X. Wang. "Quaternion-based robust attitude control for quadrotors". In: *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. 2015, pp. 920–925 (cit. on p. 27).

[20] S. Di Lucia, G. D. Tipaldi, and W. Burgard. "Attitude stabilization control of an aerial manipulator using a quaternion-based backstepping approach". In: *Mobile Robots (ECMR), 2015 European Conference on*. 2015, pp. 1–6 (cit. on p. 27).

[21] H. Bouadi, M. Bouchoucha, and M. Tadjine. "Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor". In: *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering* 1.2 (2007) (cit. on p. 27).

[22] J. Colmenares-Vazquez et al. "Integral backstepping control for trajectory tracking of a hybrid vehicle". In: *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. 2015, pp. 209–217 (cit. on pp. 27, 121).

[23] E. Reyes-Valeria et al. "LQR control for a quadrotor using unit quaternions: Modeling and simulation". In: *Electronics, Communications and Computing (CONIELECOMP), 2013 International Conference on*. 2013, pp. 172–178 (cit. on p. 27).

[24] W. Ren and R.W. Beard. "CLF-based tracking control for UAV kinematic models with saturation constraints". In: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*. Vol. 4. 2003, 3924–3929 vol.4 (cit. on p. 27).

[25] J.-F. Guerrero Castellanos et al. "Attitude Stabilization of a Quadrotor by Means of Event-Triggered Nonlinear Control". In: *Journal of Intelligent and Robotic Systems* 73 (2014). hal-00860840, pp. 123–135 (cit. on p. 27).

[26] J.-F. Guerrero-Castellanos et al. "Bounded attitude control of rigid bodies: Real-time experimentation to a quadrotor mini-helicopter". In: 19.8 (2011). hal-00568075, pp. 790–797 (cit. on p. 27).

[27] Laura E. Munoz, Omar Santos, and Pedro Castillo. "Robust nonlinear real-time control strategy to stabilize a PVTOL aircraft in crosswind". In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. 2010, pp. 1606–1611 (cit. on p. 27).

[28] Pedro Castillo, R. Lozano, and A. Dzul. "Stabilization of a mini rotorcraft with four rotors". In: *Control Systems, IEEE* 25.6 (2005), pp. 45–55 (cit. on p. 27).

[29] A. Hably and N. Marchand. "Global stabilization of a four rotor helicopter with bounded inputs". In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. 2007, pp. 129–134 (cit. on p. 27).

[30] A. Tayebi and S. McGilvray. "Attitude stabilization of a four-rotor aerial robot". In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 2. 2004, 1216–1221 Vol.2 (cit. on p. 27).

[31] Augustin Manecy et al. "X4-MaG: A Low-Cost Open-Source Micro-Quadrotor and its Linux-Based Controller". In: *International Journal of Micro Air Vehicles* 7.2 (2015), pp. 89–109. eprint: `https://doi.org/10.1260/1756-8293.7.2.89` (cit. on p. 27).

[32] C. B. Low. "A Trajectory Tracking Control Design for Fixed-Wing Unmanned Aerial Vehicles". In: *IEEE Conference on Control Applications* (Sept. 2010), pp. 2118–2123 (cit. on p. 27).

[33] M. Tahar, K. Meguenni Zemalache, A. Omari. "Control of an Under-Actuated X4-flyer using Integral Backstepping Controller". In: *PRZEGLAD ELEKTROTECHNICZNY (Electrical Review) ISSN 0033-2097, R. 87 NR 10/2011* () (cit. on p. 27).

[34] Hao Nguyen Dang, Mohamed Boutayeb, and Hugues Rafaralahy. "Trajectory-tracking control design for an under-actuated quadrotor". In: *13th European Control Conference, ECC'14*. Strasbourg, France, June 2014 (cit. on p. 27).

[35] P. Pharpatara, B. Hérissé, and Y. Bestaoui. "3-D Trajectory Planning of Aerial Vehicles Using RRT". In: *IEEE Transactions on Control Systems Technology* 25.3 (2017), pp. 1116–1123 (cit. on p. 27).

[36] C. Samson, K. Ait-Abderrahim. "Feedback control of a nonholonomic wheeled cart in Cartesian space". In: *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. 2. 1991, pp. 1136–1141 (cit. on p. 27).

[37] A. Benalia, M. Djemai, and J-P. Barbot. "Control of the kinematic car using trajectory generation and the high order sliding mode control". In: *Systems, Man and Cybernetics, 2003. IEEE International Conference on*. Vol. 3. 2003, pp. 2455–2460 (cit. on p. 27).

[38] Chuan Wang et al. "Multi-smart car control system design and research based on ZigBee". In: *Control and Decision Conference (2014 CCDC), The 26th Chinese*. 2014, pp. 1490–1494 (cit. on p. 27).

[39] In: () (cit. on p. 27).

[40] C. Aguilar-Ibanez, J.C. Martinez-Garcia, and A. Soria-Lopez. "Bounded control based on saturation functions of nonlinear under-actuated mechanical systems: The cart-pendulum system case". In: *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. 2011, pp. 1759–1764 (cit. on p. 27).

[41] Zachary Thompson Dydek. "Adaptive Control of Unmanned Aerial Systems". Ch. 3. PhD thesis. Massachusetts Institute of Technology, 2010. Chap. 3 (cit. on p. 40).

[42] A. Brezoescu, R. Lozano, and P. Castillo. "Bank to turn approach for airplane translational motion in unknown wind". In: *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*. 2013, pp. 1022–1029 (cit. on p. 40).

[43] Chang Boon Low and Quee San Ng. "A flexible virtual structure formation keeping control for fixed-wing UAVs". In: *Control and Automation (ICCA), 2011 9th IEEE International Conference on*. 2011, pp. 621–626 (cit. on p. 40).

[44] Qingji Gao, Fengfa Yue, and Dandan Hu. "Research of stability augmentation hybrid controller for quadrotor UAV". In: *Control and Decision Conference (2014 CCDC), The 26th Chinese*. 2014, pp. 5224–5229 (cit. on p. 40).

[45] H. Khebbache, M. Tadjine. "Robust Fuzzy Backstepping Sliding Mode Controller For a Quadrotor Unmanned Aerial Vehicle". In: *CEAI* 15.2 () (cit. on p. 40).

[46] K. Shojaei, A.M. Shahri, and B. Tabibian. "Adaptive-robust feedback linearizing control of a nonholonomic wheeled mobile robot". In: *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on*. 2010, pp. 497–502 (cit. on p. 40).

[47] A.-C. Brezoescu et al. "Adaptive Trajectory Following for a Fixed-Wing UAV in Presence of Crosswind". In: *Journal of Intelligent & Robotic Systems* 69.Issue 1-4 (2013), pp. 257–271 (cit. on p. 40).

[48] I. D. Landau, R. Lozano, M. M'Saad, A. Karimi. "Adaptive Control". In: Springer, 2011. Chap. 1,3 (cit. on p. 40).

[49] Y. Bouzid, H. Siguerdidjane, and Y. Bestaoui. "3D trajectory tracking control of quadrotor UAV with on-line disturbance compensation". In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. 2017, pp. 2082–2087 (cit. on p. 40).

[50] Jinpeng Yang et al. "Self-tuning PID control design for quadrotor UAV based on adaptive pole placement control". In: *Chinese Automation Congress (CAC), 2013*. 2013, pp. 233–237 (cit. on p. 40).

[51] YD Song and WC Cai. "NEW INTERMEDIATE QUATERNION BASED CONTROL OF SPACECRAFT: PART I - ALMOST GLOBAL ATTITUDE TRACKING". In: *International Journal of Innovative Computing Information and Control*. Vol. 8. 2012, pp. 7307–7319 (cit. on p. 45).

[52] WC Cai and YD Song. "NEW INTERMEDIATE QUATERNION BASED CONTROL OF SPACECRAFT: PART II - GLOBAL ATTITUDE TRACKING". In: *International Journal of Innovative Computing Information and Control*. Vol. 8. 2012, pp. 7853–7864 (cit. on p. 45).

[53] J. Cariño, H. Abaunza, and P. Castillo. "Quadrotor quaternion control". In: *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. 2015, pp. 825–831 (cit. on p. 45).

[54] H. Abaunza et al. "Quadrotor aerial manipulator based on dual quaternions". In: *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2016, pp. 152–161 (cit. on p. 45).

[55] A. Benallegue, A. Mokhtari, and L. Fridman. "Feedback linearization and high order sliding mode observer for a quadrotor UAV". In: *Variable Structure Systems, 2006. VSS'06. International Workshop on*. 2006, pp. 365–372 (cit. on p. 49).

[56] Ali Zemouche and Mohamed Boutayeb. "Observers synthesis method for a class of nonlinear discrete-time systems with extension to observer-based control". In: *17th IFAC World Congress*. Seoul, South Korea, July 2008, CDROM (cit. on p. 49).

[57] Mohamed Boutayeb et al. "A simple time-varying observer for speed estimation of UAV". In: *17th IFAC World Congress*. Seoul, South Korea, July 2008, CDROM (cit. on p. 49).

[58] K. S. Kim, K. H. Rew, and S. Kim. "Disturbance Observer for Estimating Higher Order Disturbances in Time Series Expansion". In: *IEEE Transactions on Automatic Control* 55.8 (2010), pp. 1905–1911 (cit. on p. 50).

[59] GIPSA-LAB. *MOCA Platform*. URL: `http://www.gipsa-lab.grenoble-inp.fr/recherche/plates-formes.php?id_plateforme=79` (cit. on p. 84).

[60] horizonhobby. *NanoQX*. URL: `https://www.horizonhobby.com/product/drone` (cit. on p. 87).

[61] MOCA room. *Experimental Tests*. URL: `https://drive.google.com/drive/folders/1UBFd_-OBYWLGr4dxijhVDIcfAfNhwjWV?usp=sharing` (cit. on p. 118).

[62] J. Colmenares-Vázquez et al. "Position control of a quadrotor under external constant disturbance". In: *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. 2015, pp. 180–185 (cit. on p. 121).

[63] J. Colmenares-Vázquez et al. "An intermediary quaternion-based control for trajectory following using a quadrotor". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 5965–5970 (cit. on p. 121).

[64] Pedro Castillo-García, Laura Elena Muñoz Hernandez, and Pedro García Gil. "Chapter 6 - Nonlinear Control Algorithms with Integral Action*". In: *Indoor Navigation Strategies for Aerial Autonomous Systems*. Ed. by Pedro Castillo-García et al. Butterworth-Heinemann, 2017, pp. 133 –155 (cit. on p. 121).

[65] J. J. Tellez-Guzman et al. "Velocity control of mini-UAV using a helmet system". In: *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. 2015, pp. 329–335 (cit. on p. 121).

Résumé — Ce travail porte sur la conception d'un véhicule hybride qui puisse rouler ou voler si nécessaire. Une révision des prototypes actuels est faite et ensuite il est proposé un nouveau type de véhicule avec des propriétés améliorées. Ce véhicule se constitue d'un quadrirotor avec deux roues passives sans aucun servomécanisme additionnel. Le modèle de ce véhicule a été développé et quelques algorithmes de régulation automatique ont été développés pour chaque mode d'opération : par terre ou dans l'air. Ces algorithmes ont été validés dans un premier temps via des simulations numériques. La dernière section décrit la salle MOCA et le système utilisé pour obtenir la position et l'orientation du véhicule. En outre, on a fait des expériences pour valider les stratégies de commande développées. D'autres expériences ont été fait pour valider le fait que le véhicule peut suivre des trajectoires terrestres et aériennes. Finalement, un observateur a été utilisé pour identifier le moment où le véhicule fait contact avec le sol. Ce fait nous a permis de changer du mode aérien vers le mode terrestre sans utiliser des capteurs additionnels.

Mots clés : Véchicule hybride, regulation automatique, control adaptable, drone, quadrirotor, saturation anidées, chariot, modélisation mathématique, observateur, perturbations.

Abstract — This works is focused in the conception of a hybrid vehicle with the ability to fly or roll over the ground when necessary. An revision of the present prototypes is made and then a new prototype is proposed with enhanced characteristics. The prototype consists of a quadrotor with two additional passive wheels without any other additional servomechanism. The modeling of each operation mode is exposed and then a model for the hybrid form is proposed. In the next section, some control algorithms have been developed for each operation mode: on ground or in air. These algorithms are validated via numerical simulation in the first instance. The last section describes the MOCA room and the system used for capturing the position and attitude of the vehicle. Also, experimental tests were made in order to validate the developed control strategies. It was also validated the vehicle can follows aerial and terrestrial trajectories. Finally, an observer is implemented in order to identify the moment when the vehicle touches the ground. This knowledge is used to switch from air mode to ground mode without any additional sensors.
Keywords: Hybrid vehicle, terrestrial, aerial, control, adaptive technique, UAV, quadcopter, modeling, cart, backstepping, nested saturations, disturbance observer.

GIPSA-Lab, 11 rue des Mathématiques, Grenoble Campus BP46, F-38402
SAINT MARTIN D'HERES, FRANCE