Modeling and control of a suspended load lifted by two aerial vehicles

MASSIMILIANO PANDOLFO

M.Sc. Thesis
Padova, Italia 2017

Abstract

In this thesis we discuss the problem of lifting and stabilizing a a bar-shaped slung load, held by two UAVs (Unmanned Aerial Vehicles) which are connected to it through cables. A mathematical model of the system is presented. After describing the system dynamics, we identify a set of equilibrium poses and discuss the stability of a point in this set. Then, the Gazebo simulator is used to develop and test a controller capable of stabilizing the system, rejecting disturbances, and following basic trajectories.

Contents

Acknowledgments

This thesis is the result of a semester spent abroad, at the KTH Royal Institute of Technology in Stockholm. As this experience comes to a close, it is inevitable to look back with fondness to a city and a country that have much to offer, both for the professional and the personal growth of the individual.

When it comes to thanking those who made this experience possible, the first thought must go to my parents, whose loving support accompanied me every step of the way.

I must also thank my advisor, Alessandro Chiuso, for giving me this opportunity, and my examiner, Dimos Dimarogonas, for welcoming me in Stockholm. And of course, I want to thank Pedro Pereira, my supervisor, whose advice contributed to this work.

And finally, a friendly appreciation goes out to Antonio, Matteo, Dario, Umar, Pedro and Paul, for all the good times that we shared during my stay abroad.

Chapter 1

Introduction

In this thesis, we address the problem of collaborative load lifting with aerial vehicles. Specifically, we consider a bar-shaped slung load, held by two UAVs (Unmanned Aerial Vehicles) which are connected to it through cables.

In the first part of the thesis, a mathematical model is presented. After describing the system dynamics, we identify a set of equilibrium poses and discuss the stability of a point in this set. Then, in the second part of the thesis, the Gazebo simulator is used develop and test a controller capable of stabilizing the system, rejecting disturbances, and following basic trajectories.

Motivation

In the last few years the attention towards UAVs has risen, as they are more and more viewed as a promising platform for performing various tasks, ranging from inspection to aerial manipulation. Specifically, aerial manipulation involves some sort of interaction with the environment. In order to achieve it, the vehicles are often outfitted with some sort of gripper, robotic arm or cable. Load lifting and transportation is an essential aspect of any aerial manipulation task.

In this thesis, we focus on the issue of stabilizing a slung load, using two UAVs in order to achieve a greater degree of control. The study of tethered transportation is motivated by the lower costs and lower complexity when compared to robotic arms or grippers. Moreover, there are excellent real life examples of transport by towed cables, such as emergency rescue or the delivery of a payload to an area or position impossible to reach with ground vehicles.

Literature review

The issues with the transportation of a slung load have been explored in several different publications. It has been shown that it is possible, for a point mass connected to multiple quadrotor UAVs via rigid massless links, to follow a desired trajectory while keeping the vehicles in formation [1].

The results have then also been extended to allow trajectory tracking with a rigid body as payload, still connected to an arbitrary number of aerial vehicles via rigid links [2].

When a slung load is supported by cables, additional difficulties arise since it then becomes necessary to take into consideration the tensions along the cables. This issue is discussed in [3], where different poses of the load (a rigid body) are associated to corresponding configurations of the UAVs, while respecting constraints on the tensions.

It has also been shown that an aerial vehicle connected to a cable can be modeled as a hybrid system [4–6]. In [4], the notion of a differentially-flat hybrid system is introduced, and is used to plan a trajectory for the vehicle capable of minimizing the load swing.

Methods have been devised in order to guarantee that the hybrid system never switches, thus allowing to model the cables as rigid links. A system made up of a fully actuated UAV connected to a point mass is considered in [5], where the proposed control algorithm guarantees that the cable is always under tensile forces, thus allowing to treat the system as an underactuated aerial vehicle. This approach is then extended in [6], where the point mass is connected to two aerial vehicles.

Statement of contribution

The goal of this endeavor is the stabilization of a bar shaped suspended load, connected to two multirotor UAVs by cables.

In the mathematical model, we take into consideration the odometry measurements of the vehicles and of the load,

$$\mathbf{x} = \begin{bmatrix} \mathcal{O} \\ \mathcal{O}_1 \\ \mathcal{O}_2 \end{bmatrix} .$$

where

$\mathcal{O}$ odometry of the load,

$\mathcal{O}_1$ odometry of the first UAV,

$\mathcal{O}_2$ odometry of the second UAV,

and study the dynamics of the overall system by solving the associated Newton-Euler equations.

The bar is modeled as a thin rod, and its odometry measurement $\mathcal{O}$ includes position, orientation, linear and angular velocities. The vehicles instead are treated as point masses, and thus $\mathcal{O}_1$ and $\mathcal{O}_2$ only contain position an linear velocity. The underlying assumption is that the attitude of each vehicle is handled by another layer of control.

Specifically, for each of the two UAVs, our proposed controller returns as output a vector $\mathbf{u}_i \in \mathbb{R}^3$, which represents the total upward thrust generated by the vehicle, seen in the external fixed reference frame. Therefore, $\mathbf{u}_i$ contains the information on the attitude of the vehicle; in a quadrotor, for instance, the thrust is aligned with the vertical axis of the body reference frame. We assume then that there is an inner layer of control that adjusts roll, pitch, yaw and rotor thrusts based on this information.

Our proposed control law $\mathbf{u}_{\text{UAV}}$ computes $\mathbf{u}_i$ through the combination of a PID position controller and a linear oscillation damping term responsible for rejecting disturbances and minimizing the swinging motion of the load. Note that this allows implementation with a distributed approach: we will in fact show that, for each UAV, the controller only needs information on its own odometry and on the odometry of the load,

$$\mathbf{u}_i = \mathbf{u}_{\text{UAV}}(\mathcal{O}, \mathcal{O}_i) \qquad\qquad i = 1, 2\,. \tag{1.1}$$

The code used in the simulations, including the Gazebo model of the system, is publicly available at `https://github.com/massimilianop/collaborative_load_lifting`, where additional materials, as well as some videos of the simulations, can be found.

Thesis outline

The remainder of this thesis is structured as follows. In Chapter 2, we present the mathematical model, and discuss the system dynamics. In Chapter 3, we define a chart to maps a subset of the state space, which is a manifold, in the Euclidean space. In Chapter 4, we identify a set of equilibrium configurations, and study the stability of the system through linearization. In Chapter 5, we express the system as a Gazebo model and discuss a method to test the controller in the simulator. In Chapter 6, we use the simulation results to design a controller, with an emphasis on oscillation damping. In Chapter 7, we summarize the results, and discuss directions for future research.

Chapter 2

Mathematical model

In this chapter we present the mathematical model. After a description of the system in terms of the chosen variables, the associated state space is characterized as a manifold in $\mathbb{R}^{24}$. The chapter ends with an analysis of the system dynamics.

2.1 Notation

Let $n \in \mathbb{N}$. The n-dimensional Euclidean space is denoted by $\mathbb{R}^n$. The set of unit vectors in $\mathbb{R}^n$ is denoted by

$$\mathbb{S}^{n-1} = \{\mathbf{n} \in \mathbb{R}^n : \|\mathbf{n}\| = 1\}$$

where the operator $\| \, \|$ denotes the Euclidean norm of a vector. The other commonly used operators are $\times$ to denote the cross product, and the superscript $^\intercal$ to mark the transpose of a vector or matrix. To a much lesser degree, the operator $\cdot$ is used to indicate the matrix product, but only when the absence of it may lead to ambiguity.

In general, unless explicitly stated, the following convention is used when naming variables:

$\mathbf{v} \in \mathbb{R}^n$ a lowercase bold letter refers to a column vector,

$\mathbf{M} \in \mathbb{R}^{n \times m}$ an uppercase bold letter refers to a matrix,

d, $L \in \mathbb{R}$ italic letters, both uppercase and lowercase, refer to scalar values.

The null vector in $\mathbb{R}^n$ marks an exception to this naming convention. When there is room for misinterpretation it is denoted as 0_n but otherwise, when there are no ambiguities, we drop the subscript in order to lighten the notation. For the same reason, the identity matrix is referred to as $\mathbf{I}$.

Reference frames

The reference frame that is fixed in $\mathbb{R}^3$ is denoted by xyz, where obviously x, y and z mark the names of the three axes. The standard unit vectors associated to xyz are denoted by

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} .$$

In some sections a second frame is also introduced, and it is denoted as $x'y'z'$. Matrices and vectors written with respect to the fixed coordinate frame do not present additional

superscripts (e.g. $\mathbf{v}$ and $\mathbf{M}$); when instead they are written in the coordinates of the second reference frame, they display the superscript $'$ (e.g. $\mathbf{v}'$ and $\mathbf{M}'$).

Let $\mathbf{e}_{x'}$, $\mathbf{e}_{y'}$ and $\mathbf{e}_{z'}$ denote the standard unit vectors associated to $x'y'z'$. Then, clearly, when expressed in the local coordinates they are

$$\mathbf{e}'_{x'} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{e}'_{y'} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \mathbf{e}'_{y'} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} .$$

Four-quadrant inverse tangent

The inverse trigonometric functions play a key role in describing orientations in terms of angles. Specifically, in order to compute the arctangent, we use its two-argument variant

$$\operatorname{arctan2}(x, y) = \begin{cases} \arctan(y/x) & x > 0 \\ \arctan(y/x) + \pi & x < 0\,, y > 0 \\ \arctan(y/x) - \pi & x < 0\,, y < 0 \\ +\pi/2 & x = 0\,, y > 0 \\ -\pi/2 & x = 0\,, y < 0 \end{cases}$$

since it allows for a wider range of angles than its counterpart:

$$\begin{aligned} \arctan \quad &: && \mathbb{R} \quad \longrightarrow \quad \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \\ \operatorname{arctan2} \quad &: \quad \mathbb{R}^2 \backslash \{(x, y) : x \leq 0\,, \;\; y = 0\} \quad \longrightarrow \quad (-\pi, \pi) \;. \end{aligned}$$

Note that with this domain we ensure that arctan2 is continuous, which is required when we define the chart in Chapter 3. In fact, at $y = 0$ there is a discontinuity for all $x < 0$, as is shown in Figure 2.1.

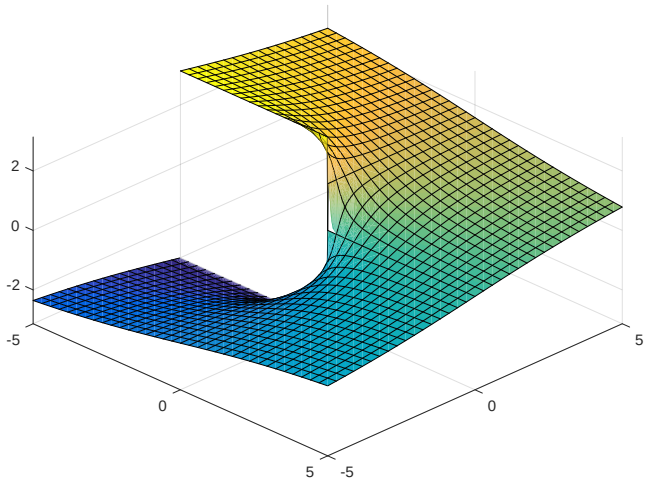


Figure 2.1: Plot of the function arctan2

Skew-symmetric matrix and cross product

In some parts of this thesis, the cross product is expressed as a matrix product

$$\mathbf{a} \times \mathbf{b} = \mathbf{S}(\mathbf{a}) \cdot \mathbf{b}$$

where the skew-symmetric matrix $\mathbf{S}(\mathbf{a})$ is defined as

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \qquad \text{with} \qquad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} .$$

2.2 Description of the system

The system represented in Figure 2.2 features two UAVs of masses m_1 and m_2 connected to a bar-shaped load of mass m_L through two cables of length L_1 and L_2. The relevant system variables, all expressed in an external fixed reference frame xyz (also referred to as global or world reference frame), are denoted as

$\mathbf{p}$ position of the center of mass of the load

$\mathbf{n}_B$ unit vector associated to the orientation of the load

$\mathbf{v}$ linear velocity of $\mathbf{p}$

$\boldsymbol{\omega}_B$ angular velocity of $\mathbf{p}$

$\mathbf{p}_1$ position of the first UAV

$\mathbf{v}_1$ linear velocity of $\mathbf{p}_1$

$\mathbf{p}_2$ position of the second UAV

$\mathbf{v}_2$ linear velocity of $\mathbf{p}_2$.

In order to simplify the analysis, the load is modeled as a slender rod, with a thickness that is negligible when compared to its length ℓ.

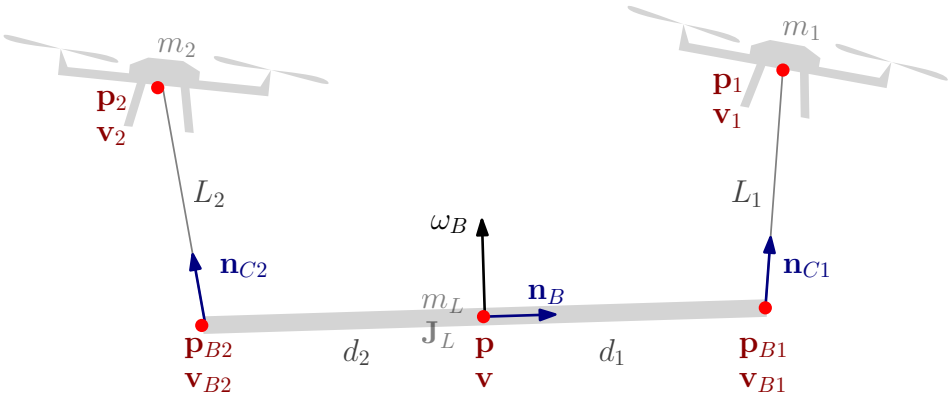


Figure 2.2: Representation of the system

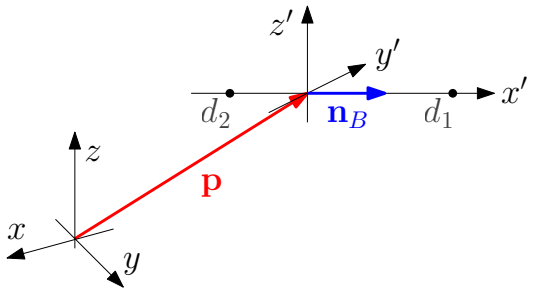


Figure 2.3: Diagram showing the different reference frames

Orientation of the load

Let us consider a load based reference frame $x'y'z'$ centered in $\mathbf{p}$ and such that the x' axis is aligned with the unit vector $\mathbf{n}_B$, as in Figure 2.3.

By definition the orientation of $x'y'z'$ is characterized by the coordinates of its standard unit vectors $\mathbf{e}_{x'}$, $\mathbf{e}_{y'}$ and $\mathbf{e}_{z'}$ expressed with respect to the global reference frame xyz:

$$\mathbf{R} = \begin{bmatrix} \mathbf{e}_{x'} & \mathbf{e}_{y'} & \mathbf{e}_{z'} \end{bmatrix} = \begin{bmatrix} \mathbf{n}_B & \mathbf{e}_{y'} & \mathbf{e}_{z'} \end{bmatrix} . \tag{2.1}$$

Since the load is modeled as a one-dimensional object, its orientation is associated with the direction of the local axis x' and can therefore be described by the unit vector $\mathbf{n}_B$.

Inertia of the load

The inertia tensor of the slender rod, expressed in the local reference frame $x'y'z'$, is the constant matrix

$$\mathbf{J}'_L = J_\ell \begin{bmatrix} \varepsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \frac{1}{12} m_L \ell^2 \begin{bmatrix} \varepsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

where $0 < \varepsilon \ll 1$ ensures that $\mathbf{J}'_L$ is invertible. The matrix can also be expressed in the global reference frame xyz:

$$\mathbf{J}_L = \mathbf{R}\, \mathbf{J}'_L\, \mathbf{R}^\intercal \tag{2.3}$$

where $\mathbf{R}$ is the rotation matrix defined in (2.1).

Anchoring points

Consider the diagram in Figure 2.3. Since the dimensions of the object along the y' and z' axes are assumed to be negligible, then clearly every point of the body can be described just by its coordinate along the x' axis.

In particular, in the local reference frame, the anchoring points between load and cables are denoted by their coordinates d_1 and d_2, according to

$$\mathbf{p}'_{Bi} = d_i\, \mathbf{e}'_{x'} \qquad i = 1, 2\,.$$

From this characterization, it is trivial to express the anchoring points in the global reference frame:

$$\mathbf{p}_{Bi} = \mathbf{p} + d_i \mathbf{n}_B \qquad i = 1, 2\,. \tag{2.4}$$

The linear velocities associated to the anchoring points are easily obtained by computing the time derivative of (2.4):

$$\mathbf{v}_{Bi} = \mathbf{v} + d_i \boldsymbol{\omega}_B \times \mathbf{n}_B \qquad i = 1, 2\,. \tag{2.5}$$

2.3 State space

The overall state of the system takes into account all the system variables, and is defined as the 24-dimensional vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{n}_B \\ \boldsymbol{\omega}_B \\ \mathbf{p}_1 \\ \mathbf{v}_1 \\ \mathbf{p}_2 \\ \mathbf{v}_2 \end{bmatrix} \in \left(\mathbb{R}^3\right)^8 . \tag{2.6}$$

The control input is described by the pair

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \in \left(\mathbb{R}^3\right)^2 \tag{2.7}$$

where

$\mathbf{u}_1 \in \mathbb{R}^3$ is the total upward force generated by the first UAV

$\mathbf{u}_2 \in \mathbb{R}^3$ is the total upward force generated by the second UAV.

Note that the full pose of the load appears in (2.6), while instead the UAVs are characterized only by information related to their position. As stated in the introduction, the underlying assumption is that the internal controller of the UAVs handles their orientation, adjusting the roll, pitch, yaw and rotor thrusts of each vehicle in order to obtain the desired force $\mathbf{u}_i$.

When defining the state space, it is fundamental to study the various restrictions acting upon the elements stacked in (2.6). As discussed in [5, 6], the quadrotors-load system can be modeled as a hybrid system, exhibiting two behaviors. When neither of the cables are under tension the load behaves like a free falling object, while the UAVs behave as standard unconstrained aerial vehicles. When instead one or both cables are under tension, one or two kinematic constraints are introduced in the system, stating that the distance between $\mathbf{p}_i$ and $\mathbf{p}_{Bi}$ is constant and equal to the length L_i of the taut cable.

In this work, the system is analyzed under the assumption that both cables are always taut, and therefore the cables are modeled as rigid links, each introducing a kinematic constraint in the state space. In light of these observations, the state space shapes up to be a manifold $\Omega \subset \left(\mathbb{R}^3\right)^8$, and the following restrictions hold:

$$\begin{aligned}
\Omega = \big\{ \mathbf{x} \text{ as in (2.6)} \,:\, & \|\mathbf{n}_B\| = 1 && && \text{(2.8a)} \\
& \mathbf{n}_B \perp \boldsymbol{\omega}_B && && \text{(2.8b)} \\
& \|\mathbf{p}_i - \mathbf{p}_{Bi}\| = L_i && i = 1, 2 && \text{(2.8c)} \\
& (\mathbf{p}_i - \mathbf{p}_{Bi}) \perp (\mathbf{v}_i - \mathbf{v}_{Bi}) && i = 1, 2 \big\}\,. && \text{(2.8d)}
\end{aligned}$$

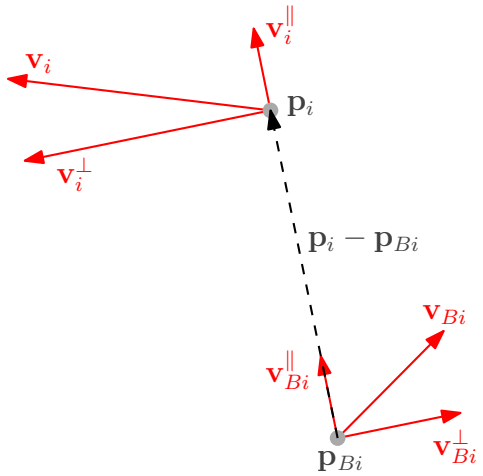


Figure 2.4: Graphical representation of restriction (2.8d)

Condition (2.8b) is a consequence on the implicit assumption that the bar does not spin around itself. It is actually possible to prove that if the vector field describing the system dynamics belongs to the tangent space, then the component of $\boldsymbol{\omega}_B$ parallel to $\mathbf{n}_B$ is constant and equal to its initial value, which can be assumed to be zero.

Restrictions (2.8c) and (2.8d) together enforce the condition that both cables behave as rigid links.

In particular, (2.8d) states that if $\mathbf{p}_i$ and $\mathbf{p}_{Bi}$ move then their distance remains constant. This concept is clarified in Figure 2.4: let the velocities be decomposed as

$$\begin{aligned}\mathbf{v}_i &= \mathbf{v}_i^{\parallel} + \mathbf{v}_i^{\perp} \\ \mathbf{v}_{Bi} &= \mathbf{v}_{Bi}^{\parallel} + \mathbf{v}_{Bi}^{\perp}\end{aligned}$$

where $\mathbf{v}_i^{\parallel}$ and $\mathbf{v}_{Bi}^{\parallel}$ are parallel to $\mathbf{p}_i - \mathbf{p}_{Bi}$, while $\mathbf{v}_i^{\perp}$ and $\mathbf{v}_{Bi}^{\perp}$ are perpendicular to $\mathbf{p}_i - \mathbf{p}_{Bi}$. Then it is easy to see that

$$\text{(2.8d) holds} \iff \mathbf{v}_i^{\parallel} = \mathbf{v}_{Bi}^{\parallel}\,,$$

which means that $\|\mathbf{p}_i - \mathbf{p}_{Bi}\|$ remains constant.

Note that if the cables are considered rigid links, then they can each be associated to a unit vector:

$$\mathbf{n}_{Ci} = \frac{\mathbf{p}_i - \mathbf{p}_{Bi}}{\|\mathbf{p}_i - \mathbf{p}_{Bi}\|} = \frac{\mathbf{p}_i - \mathbf{p}_{Bi}}{L_i} \qquad i = 1, 2\,. \tag{2.9}$$

Using (2.9) it is possible to express the relation between load and UAV positions for states $\mathbf{x} \in \Omega$:

$$\mathbf{p}_i = \mathbf{p} + d_i \mathbf{n}_B + L_i \mathbf{n}_{Ci} \qquad i = 1, 2\,. \tag{2.10}$$

2.4 Tangent space

In order to derive the tangent space of Ω at $\mathbf{x} \in \Omega$, it is helpful to express the restrictions (2.8) on the elements in Ω as scalar products:

$$\begin{aligned}
&\mathbf{n}_B^\intercal \, \mathbf{n}_B = 1 && && \text{(2.8a revisited)}\\
&\mathbf{n}_B^\intercal \, \boldsymbol{\omega}_B = 0 && && \text{(2.8b revisited)}\\
&(\mathbf{p}_i - \mathbf{p}_{Bi})^\intercal \, (\mathbf{p}_i - \mathbf{p}_{Bi}) = L_i^2 && i = 1,2 && \text{(2.8c revisited)}\\
&(\mathbf{v}_i - \mathbf{v}_{Bi})^\intercal \, (\mathbf{p}_i - \mathbf{p}_{Bi}) = 0 && i = 1,2\,. && \text{(2.8d revisited)}
\end{aligned}$$

This way, the restrictions on the elements of the tangent space can be computed as time derivatives of the scalar products above. The tangent space of Ω at $\mathbf{x} \in \Omega$ is thus defined as the set

$$\begin{aligned}
T_{\mathbf{x}}\Omega = \big\{ \delta\mathbf{x} \in \mathbb{R}^{24} : \; & (\delta\mathbf{n}_B)^\intercal \, \mathbf{n}_B = 0 && && \text{(2.11a)}\\
& (\delta\boldsymbol{\omega}_B)^\intercal \, \mathbf{n}_B + (\delta\mathbf{n}_B)^\intercal \, \boldsymbol{\omega}_B = 0 && && \text{(2.11b)}\\
& (\delta\mathbf{p}_i - \delta\mathbf{p}_{Bi})^\intercal \, (\mathbf{p}_i - \mathbf{p}_{Bi}) = 0 && i = 1,2 && \text{(2.11c)}\\
& (\delta\mathbf{v}_i - \delta\mathbf{v}_{Bi})^\intercal (\mathbf{p}_i - \mathbf{p}_{Bi}) + (\mathbf{v}_i - \mathbf{v}_{Bi})^\intercal (\delta\mathbf{p}_i - \delta\mathbf{p}_{Bi}) = 0 && i = 1,2 \big\} && \text{(2.11d)}
\end{aligned}$$

where each element in $T_{\mathbf{x}}\Omega$ is a stacked vector with the following structure

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p} \\ \delta\mathbf{v} \\ \delta\mathbf{n}_B \\ \delta\boldsymbol{\omega}_B \\ \delta\mathbf{p}_1 \\ \delta\mathbf{v}_1 \\ \delta\mathbf{p}_2 \\ \delta\mathbf{v}_2 \end{bmatrix}$$

and where the auxiliary values $\delta\mathbf{p}_{Bi}$ and $\delta\mathbf{v}_{Bi}$ are defined according to

$$\begin{aligned}
\delta\mathbf{p}_{Bi} =& \delta\mathbf{p} + d_i \, \delta\mathbf{n}_B && i = 1,2\\
\delta\mathbf{v}_{Bi} =& \delta\mathbf{v} + d_i \, (\delta\boldsymbol{\omega}_B \times \mathbf{n}_B + \boldsymbol{\omega}_B \times \delta\mathbf{n}_B) && i = 1,2\,.
\end{aligned}$$

2.5 Dynamic equations

The dynamic equations are associated to the time derivative of the state. Let $\mathbf{x} \in \Omega$ and $\mathbf{u} \in \left(\mathbb{R}^3\right)^2$, then the open loop vector field is defined as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{n}}_B \\ \dot{\boldsymbol{\omega}}_B \\ \dot{\mathbf{p}}_1 \\ \dot{\mathbf{v}}_1 \\ \dot{\mathbf{p}}_2 \\ \dot{\mathbf{v}}_2 \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u})\,, \tag{2.12}$$

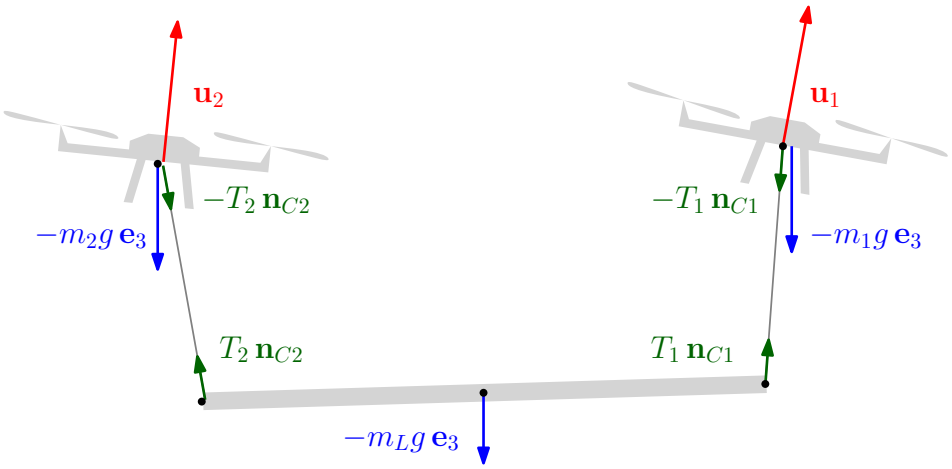


Figure 2.5: Representation of the forces acting on the system

where, by the definition of time derivative

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{n}}_B = \boldsymbol{\omega}_B \times \mathbf{n}_B \\ \dot{\mathbf{p}}_i = \mathbf{v}_i \qquad\qquad i = 1, 2 \end{cases}$$

and where the accelerations $\dot{\mathbf{v}}$, $\dot{\boldsymbol{\omega}}_B$, $\dot{\mathbf{v}}_1$ and $\dot{\mathbf{v}}_2$ must be such that, based on the schematic representation in Figure 2.5, the following system of equations holds:

$$\begin{cases} m_1\dot{\mathbf{v}}_1 = \mathbf{u}_1 - T_1\,\mathbf{n}_{C1} - m_1 g\,\mathbf{e}_3 & \text{(2.13a)} \\ m_2\dot{\mathbf{v}}_2 = \mathbf{u}_2 - T_2\,\mathbf{n}_{C2} - m_2 g\,\mathbf{e}_3 & \text{(2.13b)} \\ m_L\dot{\mathbf{v}} = T_1\,\mathbf{n}_{C1} + T_2\,\mathbf{n}_{C2} - m_L g\,\mathbf{e}_3 & \text{(2.13c)} \\ \dfrac{d\,(\mathbf{J}_L\boldsymbol{\omega}_B)}{dt} = d_1\,\mathbf{n}_B \times T_1\,\mathbf{n}_{C1} + d_2\,\mathbf{n}_B \times T_2\,\mathbf{n}_{C2}\,. & \text{(2.13d)} \end{cases}$$

Notice how the dynamics of the load, being a rigid body, are described by the Newton-Euler equations (2.13c) and (2.13d).

In particular, (2.13d) can be simplified by taking into account the structure of the inertia tensor. Let us rewrite it as

$$\frac{d}{dt}\,(\mathbf{J}_L\boldsymbol{\omega}_B) = \sum \tau \qquad \text{(2.13d revisited)}$$

where the term on the right side of the equation is used to indicate the sum of all torques. Then, by (2.3) the time derivative yields

$$\begin{aligned} \frac{d}{dt}\,(\mathbf{J}_L\boldsymbol{\omega}_B) &= \frac{d}{dt}\,(\mathbf{R}\,\mathbf{J}'_L\,\mathbf{R}^\intercal\boldsymbol{\omega}_B) \\ &= \mathbf{S}(\boldsymbol{\omega}_B)\mathbf{R}\,\mathbf{J}'_L\,\mathbf{R}^\intercal\boldsymbol{\omega}_B - \cancel{\mathbf{R}\,\mathbf{J}'_L\,\mathbf{R}^\intercal\,\mathbf{S}(\boldsymbol{\omega}_B)\boldsymbol{\omega}_B} + \mathbf{R}\,\mathbf{J}'_L\,\mathbf{R}^\intercal\dot{\boldsymbol{\omega}}_B \end{aligned} \qquad \text{(2.14)}$$

where the cancellation is possible since $\mathbf{S}(\boldsymbol{\omega}_B)\cdot\boldsymbol{\omega}_B$ is equal to the null vector.

Recall that the inertia tensor $\mathbf{J}'_L$, defined in (2.2), is a diagonal matrix and can therefore be rewritten as

$$\begin{aligned}\mathbf{J}'_L &= J_\ell \begin{bmatrix} \varepsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= J_\ell \, \mathbf{I} + (\varepsilon - 1)\, J_\ell \, \mathbf{e}_1 \mathbf{e}_1^\intercal \,. \end{aligned} \tag{2.15}$$

Then, by plugging (2.15) in equation (2.14) we obtain

$$\begin{aligned}\frac{d\,(\mathbf{J}_L \boldsymbol{\omega}_B)}{dt} &= \cancel{J_\ell \mathbf{S}(\boldsymbol{\omega}_B)\boldsymbol{\omega}_B} + (\varepsilon - 1)\, J_\ell \mathbf{S}(\boldsymbol{\omega}_B)\mathbf{R}\mathbf{e}_1\mathbf{e}_1^\intercal\mathbf{R}^\intercal\boldsymbol{\omega}_B + J_\ell\,\dot{\boldsymbol{\omega}}_B + (\varepsilon - 1)\, J_\ell \mathbf{R}\mathbf{e}_1\mathbf{e}_1^\intercal\mathbf{R}^\intercal\dot{\boldsymbol{\omega}}_B \\ &= J_\ell\,\dot{\boldsymbol{\omega}}_B + (\varepsilon - 1)\, J_\ell \left(\cancel{\dot{\mathbf{n}}_B \mathbf{n}_B^\intercal \boldsymbol{\omega}_B} + \mathbf{n}_B\,\mathbf{n}_B^\intercal\,\dot{\boldsymbol{\omega}}_B \right)\end{aligned}$$

which allows to rewrite (2.13d) as

$$J_\ell\,\dot{\boldsymbol{\omega}}_B + (\varepsilon - 1)\, J_\ell\,\mathbf{n}_B\,\mathbf{n}_B^\intercal\,\dot{\boldsymbol{\omega}}_B = d_1\,\mathbf{n}_B \times T_1\,\mathbf{n}_{C1} + d_2\,\mathbf{n}_B \times T_2\,\mathbf{n}_{C2}\,. \tag{2.16}$$

Without additional knowledge on the vector field, the values T_1 and T_2 of the tensions along the cables that appear in (2.13) and (2.16) are unknown. Their explicit expressions $T_1(\mathbf{x}, \mathbf{u})$ and $T_2(\mathbf{x}, \mathbf{u})$ can be computed by guaranteeing that

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \in T_{\mathbf{x}}\Omega \qquad \forall\, \mathbf{x} \in \Omega\,,\ \forall\, \mathbf{u} \in \left(\mathbb{R}^3\right)^2\,. \tag{2.17}$$

This is done in two steps. First we show that if $\dot{\mathbf{x}} \in T_{\mathbf{x}}\Omega$ then each of the tensions corresponds to a well defined and specific function, $T_1 = T_1(\mathbf{x}, \mathbf{u})$ and $T_2 = T_2(\mathbf{x}, \mathbf{u})$; second, we verify that the vector field $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ associated to those particular tension functions indeed belongs to $T_{\mathbf{x}}\Omega$ for all values of $\mathbf{x} \in \Omega$ and $\mathbf{u} \in \left(\mathbb{R}^3\right)^2$.

Step 1: Tension functions

If the vector field belongs to $T_{\mathbf{x}}\Omega$, then $\dot{\mathbf{x}}$ as it is defined in (2.12) must satisfy the constraints in (2.11), which take the form

$$\dot{\mathbf{n}}_B^\intercal\,\mathbf{n}_B = 0 \tag{2.18a}$$

$$\dot{\boldsymbol{\omega}}_B^\intercal\,\mathbf{n}_B + \cancel{\boldsymbol{\omega}_B^\intercal\,\dot{\mathbf{n}}_B} = 0 \tag{2.18b}$$

$$(\mathbf{v}_i - \mathbf{v}_{Bi})^\intercal\,(\mathbf{p}_i - \mathbf{p}_{Bi}) = 0 \qquad i = 1, 2 \tag{2.18c}$$

$$(\dot{\mathbf{v}}_i - \dot{\mathbf{v}}_{Bi})^\intercal\,(\mathbf{p}_i - \mathbf{p}_{Bi}) + \|\mathbf{v}_i - \mathbf{v}_{Bi}\|^2 = 0 \qquad i = 1, 2\,, \tag{2.18d}$$

where the scalar product $\boldsymbol{\omega}_B^\intercal\,\dot{\mathbf{n}}_B$ can be canceled because $\boldsymbol{\omega}_B \perp \dot{\mathbf{n}}_B$, and where the accelerations $\dot{\mathbf{v}}_{B1}$ and $\dot{\mathbf{v}}_{B2}$ are obviously computed as time derivatives of (2.5):

$$\dot{\mathbf{v}}_{Bi} = \dot{\mathbf{v}} + d_i\dot{\boldsymbol{\omega}}_B \times \mathbf{n}_B - d_i\,\|\boldsymbol{\omega}_B\|^2\,\mathbf{n}_B \qquad i = 1, 2\,. \tag{2.19}$$

Note that (2.18a) and (2.18c) do not add any new information, and therefore we will focus on restrictions (2.18b) and (2.18d). Starting with (2.18b), if we combine it with (2.16) then the expression of the torques in the Newton-Euler equations is further simplified:

$$\dot{\boldsymbol{\omega}}_B = \frac{d_1}{J_\ell}\,\mathbf{n}_B \times T_1\,\mathbf{n}_{C1} + \frac{d_2}{J_\ell}\,\mathbf{n}_B \times T_2\,\mathbf{n}_{C2}\,. \tag{2.20}$$

Moving on to (2.18d), if we plug (2.19) in it, then the constraint can be rewritten as

$$\left(\dot{\mathbf{v}} - \dot{\mathbf{v}}_i + d_i\,\dot{\boldsymbol{\omega}}_B \times \mathbf{n}_B - d_i\,\|\boldsymbol{\omega}_B\|^2\,\mathbf{n}_B\right)^\intercal L_i\,\mathbf{n}_{Ci} = \|\mathbf{v}_i - \mathbf{v}_{Bi}\|^2 \qquad i = 1, 2 \tag{2.21}$$

which, trough a few simple algebraic steps, can be translated into the system

$$\begin{cases} \mathbf{n}_{C1}^\intercal\,(\dot{\mathbf{v}} - \dot{\mathbf{v}}_1) + d_1\,\mathbf{n}_{C1}^\intercal\,(\dot{\boldsymbol{\omega}}_B \times \mathbf{n}_B) = \dfrac{1}{L_1}\,\|\mathbf{v}_1 - \mathbf{v}_{B1}\|^2 + d_1\,\|\boldsymbol{\omega}_B\|^2\,\mathbf{n}_{C1}^\intercal\mathbf{n}_B & \text{(2.22a)} \\ \mathbf{n}_{C2}^\intercal\,(\dot{\mathbf{v}} - \dot{\mathbf{v}}_2) + d_2\,\mathbf{n}_{C2}^\intercal\,(\dot{\boldsymbol{\omega}}_B \times \mathbf{n}_B) = \dfrac{1}{L_2}\,\|\mathbf{v}_2 - \mathbf{v}_{B2}\|^2 + d_2\,\|\boldsymbol{\omega}_B\|^2\,\mathbf{n}_{C2}^\intercal\mathbf{n}_B & \text{(2.22b)} \end{cases}$$

The idea now is to express the terms on the left hand side of (2.22) in terms of the tensions T_1 and T_2. In particular, the first addendum $\mathbf{n}_{Ci}^\intercal\,(\dot{\mathbf{v}} - \dot{\mathbf{v}}_i)$ can be rewritten by considering the difference between (2.13c) and (2.13a) for the first UAV, and the difference between (2.13c) and (2.13b) for the second one, leading to:

$$\mathbf{n}_{C1}^\intercal\,(\dot{\mathbf{v}} - \dot{\mathbf{v}}_1) = T_1\left(\frac{1}{m_L} + \frac{1}{m_1}\right)\mathbf{n}_{C1}^\intercal\mathbf{n}_{C1} + T_2\frac{1}{m_L}\,\mathbf{n}_{C1}^\intercal\mathbf{n}_{C2} - \frac{1}{m_1}\,\mathbf{n}_{C1}^\intercal\mathbf{u}_1 \tag{2.23a}$$

$$\mathbf{n}_{C2}^\intercal\,(\dot{\mathbf{v}} - \dot{\mathbf{v}}_2) = T_1\frac{1}{m_L}\,\mathbf{n}_{C2}^\intercal\mathbf{n}_{C1} + T_2\left(\frac{1}{m_L} + \frac{1}{m_2}\right)\mathbf{n}_{C2}^\intercal\mathbf{n}_{C2} - \frac{1}{m_2}\,\mathbf{n}_{C2}^\intercal\mathbf{u}_2\,. \tag{2.23b}$$

Similarly, the second addendum $\mathbf{n}_{Ci}^\intercal\,(\dot{\boldsymbol{\omega}}_B \times \mathbf{n}_B)$ can be expressed in terms of the tensions thanks to (2.20), leading to

$$\mathbf{n}_{C1}^\intercal\,(\dot{\boldsymbol{\omega}}_B \times \mathbf{n}_B) = (\mathbf{n}_B \times \mathbf{n}_{C1})^\intercal\,(\mathbf{n}_B \times \mathbf{n}_{C1})\,\frac{d_1}{J_\ell}\,T_1 + (\mathbf{n}_B \times \mathbf{n}_{C1})^\intercal\,(\mathbf{n}_B \times \mathbf{n}_{C2})\,\frac{d_2}{J_\ell}\,T_2 \tag{2.24a}$$

$$\mathbf{n}_{C2}^\intercal\,(\dot{\boldsymbol{\omega}}_B \times \mathbf{n}_B) = (\mathbf{n}_B \times \mathbf{n}_{C2})^\intercal\,(\mathbf{n}_B \times \mathbf{n}_{C1})\,\frac{d_1}{J_\ell}\,T_1 + (\mathbf{n}_B \times \mathbf{n}_{C2})^\intercal\,(\mathbf{n}_B \times \mathbf{n}_{C2})\,\frac{d_2}{J_\ell}\,T_2 \tag{2.24b}$$

The result of plugging (2.23) and (2.24) in (2.22) is a linear system in the two variables T_1 and T_2. In vector form

$$\mathbf{D}(\mathbf{x}) \cdot \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \mathbf{d}(\mathbf{x}, \mathbf{u})$$

with

$$\mathbf{D}(\mathbf{x}) = \begin{bmatrix} \frac{m_L}{m_1} & 0 \\ 0 & \frac{m_L}{m_2} \end{bmatrix} + \begin{bmatrix} 1 & \mathbf{n}_{C1}^\intercal\mathbf{n}_{C2} \\ \mathbf{n}_{C2}^\intercal\mathbf{n}_{C1} & 1 \end{bmatrix} + \begin{bmatrix} d_1(\mathbf{n}_B \times \mathbf{n}_{C1})^\intercal \\ d_2(\mathbf{n}_B \times \mathbf{n}_{C2})^\intercal \end{bmatrix} \frac{m_L}{J_\ell} \begin{bmatrix} d_1\mathbf{n}_B \times \mathbf{n}_{C1} & d_2\mathbf{n}_B \times \mathbf{n}_{C2} \end{bmatrix}$$

$$\mathbf{d}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \frac{m_L}{m_1}\,\mathbf{n}_{C1}^\intercal\mathbf{u}_1 + \frac{m_L}{L_1}\,\|\mathbf{v}_1 - \mathbf{v}_{B1}\|^2 + m_L d_1\,\|\boldsymbol{\omega}_B\|^2\,\mathbf{n}_{C1}^\intercal\mathbf{n}_B \\ \frac{m_L}{m_2}\,\mathbf{n}_{C2}^\intercal\mathbf{u}_2 + \frac{m_L}{L_2}\,\|\mathbf{v}_2 - \mathbf{v}_{B2}\|^2 + m_L d_2\,\|\boldsymbol{\omega}_B\|^2\,\mathbf{n}_{C2}^\intercal\mathbf{n}_B \end{bmatrix}.$$

The matrix function $\mathbf{D}(\mathbf{x})$ is symmetric positive definite for $\mathbf{x} \in \Omega$. Therefore, the explicit form of the tension functions $T_1(\mathbf{x}, \mathbf{u})$ and $T_2(\mathbf{x}, \mathbf{u})$ is given by

$$\begin{bmatrix} T_1(\mathbf{x}, \mathbf{u}) \\ T_2(\mathbf{x}, \mathbf{u}) \end{bmatrix} = \mathbf{D}(\mathbf{x})^{-1} \cdot \mathbf{d}(\mathbf{x}, \mathbf{u}) \tag{2.25}$$

The resulting vector field is

$$\mathbf{f}(\mathbf{x},\mathbf{u}) = \begin{bmatrix} \mathbf{v} \\ {}^{1}\!/_{m_L}\, T_1(\mathbf{x},\mathbf{u})\, \mathbf{n}_{C1} + {}^{1}\!/_{m_L}\, T_2(\mathbf{x},\mathbf{u})\, \mathbf{n}_{C2} - g\, \mathbf{e}_3 \\ \boldsymbol{\omega}_B \times \mathbf{n}_B \\ {}^{d_1}\!/_{J_\ell}\, T_1(\mathbf{x},\mathbf{u})\, \mathbf{n}_B \times \mathbf{n}_{C1} + {}^{d_2}\!/_{J_\ell}\, T_2(\mathbf{x},\mathbf{u})\, \mathbf{n}_B \times \mathbf{n}_{C2} \\ \mathbf{v}_1 \\ {}^{1}\!/_{m_1}\, \mathbf{u}_1 - {}^{1}\!/_{m_1}\, T_1(\mathbf{x},\mathbf{u})\, \mathbf{n}_{C1} - g\, \mathbf{e}_3 \\ \mathbf{v}_2 \\ {}^{1}\!/_{m_2}\, \mathbf{u}_2 - {}^{1}\!/_{m_2}\, T_2(\mathbf{x},\mathbf{u})\, \mathbf{n}_{C2} - g\, \mathbf{e}_3 \end{bmatrix} \tag{2.26}$$

where $T_1(\mathbf{x},\mathbf{u})$ and $T_2(\mathbf{x},\mathbf{u})$ are the functions defined in (2.25).

Step 2: Proof that the vector field belongs to the tangent space

Let the vector field $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$ be defined according to (2.26). In order for $\dot{\mathbf{x}}$ to belong to $T_{\mathbf{x}}\Omega$, it needs to satisfy the constraints specified in (2.18):

$$\dot{\mathbf{n}}_B^{\intercal}\, \mathbf{n}_B = 0 \qquad \text{(2.18a revisited)}$$

$$\dot{\boldsymbol{\omega}}_B^{\intercal}\, \mathbf{n}_B = 0 \qquad \text{(2.18b revisited)}$$

$$(\mathbf{v}_i - \mathbf{v}_{Bi})^{\intercal}\, (\mathbf{p}_i - \mathbf{p}_{Bi}) = 0 \qquad i = 1,2 \qquad \text{(2.18c revisited)}$$

$$(\dot{\mathbf{v}}_i - \dot{\mathbf{v}}_{Bi})^{\intercal}\, (\mathbf{p}_i - \mathbf{p}_{Bi}) + \|\mathbf{v}_i - \mathbf{v}_{Bi}\|^2 = 0 \qquad i = 1,2\,. \qquad \text{(2.18d revisited)}$$

By definition, a unit vector and its time derivative are orthogonal, so clearly (2.18a) holds. It is equally trivial to prove that (2.18c) holds as long as $\mathbf{x} \in \Omega$, since it is the same as restriction (2.8d) detailed in Section §2.3. Constraint (2.18b) holds since

$$\dot{\boldsymbol{\omega}}_B = \frac{d_1}{J_\ell}\, T_1(\mathbf{x},\mathbf{u})\, \mathbf{n}_B \times \mathbf{n}_{C1} + \frac{d_2}{J_\ell}\, T_2(\mathbf{x},\mathbf{u})\, \mathbf{n}_B \times \mathbf{n}_{C2}$$

and by definition

$$(\mathbf{n}_B \times \mathbf{n}_{Ci}) \perp \mathbf{n}_B \qquad i = 1,2\,.$$

And lastly (2.18d) holds because the tension functions that appear in (2.26) were determined by solving (2.18d) in the variables T_1 and T_2.

Chapter 3

Topology of the state space

In this chapter we introduce an alternative representation of the mathematical model. In particular, we show how an appropriate parametrization of the unit vectors allows to define a chart that maps a subset of the manifold $\Omega \subset \mathbb{R}^{24}$ in a subset of $\mathbb{R}^{18}$.

3.1 Local description of the manifold

The state space $\Omega \subset \mathbb{R}^{24}$ is an 18-dimensional manifold. Thus, it is possible to describe it with an atlas, a set of homeomorphisms that each map a region of Ω to a subset of $\mathbb{R}^{18}$.

As a preliminary step, notice that thanks to the constraints enforced by (2.8c) and (2.8d) the position and linear velocity of the UAVs that appear in

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{n}_B \\ \boldsymbol{\omega}_B \\ \mathbf{p}_1 \\ \mathbf{v}_1 \\ \mathbf{p}_2 \\ \mathbf{v}_2 \end{bmatrix} \in \Omega \subset \left(\mathbb{R}^3\right)^8 \tag{3.1}$$

can be expressed in terms of the respective cable's unit vector and its angular velocity:

$$\mathbf{p}_i = \mathbf{p} + d_i\, \mathbf{n}_B + L_i\, \mathbf{n}_{Ci} \qquad i = 1, 2 \tag{3.2}$$

$$\mathbf{v}_i = \mathbf{v} + d_i\, \boldsymbol{\omega}_B \times \mathbf{n}_B + L_i\, \boldsymbol{\omega}_{Ci} \times \mathbf{n}_{Ci} \qquad i = 1, 2 \tag{3.3}$$

which hints at a one to one correspondence

$$(\mathbf{p}_i\,,\, \mathbf{v}_i) \longleftrightarrow (\mathbf{n}_{Ci}\,,\, \boldsymbol{\omega}_{Ci}) \qquad i = 1, 2$$

between a vehicle and the direction of the respective cable. Indeed, if the state were to be

expressed as

$$\mathbf{y} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{n}_B \\ \boldsymbol{\omega}_B \\ \mathbf{n}_{C1} \\ \boldsymbol{\omega}_{C1} \\ \mathbf{n}_{C2} \\ \boldsymbol{\omega}_{C2} \end{bmatrix} \in \left(\mathbb{R}^3\right)^8 \tag{3.4}$$

instead of (3.1), the corresponding state space

$$\Omega_{\mathbf{y}} = \big\{ \mathbf{y} \text{ as in (3.4)} : \|\mathbf{n}_B\| = 1 \tag{3.5a}$$

$$\mathbf{n}_B \perp \boldsymbol{\omega}_B \tag{3.5b}$$

$$\|\mathbf{n}_{Ci}\| = 1 \qquad i = 1, 2 \tag{3.5c}$$

$$\mathbf{n}_{Ci} \perp \boldsymbol{\omega}_{Ci} \qquad i = 1, 2 \big\} \tag{3.5d}$$

would be a manifold conceptually similar to Ω, to the point that the two representations are easily interchangeable.

Therefore the idea is to choose an appropriate parametrization, so that a unit vector and its angular velocity can replaced by a pair of Euler angles and their time derivatives. Note that in this thesis work, rather then characterizing the whole atlas, we limit ourselves to defining just one chart, providing a local description of the state space in the neighborhood of a certain point

$$\begin{bmatrix} 0_3 \\ 0_3 \\ \mathbf{e}_1 \\ 0_3 \\ d_1\mathbf{e}_1 + L_1\mathbf{e}_3 \\ 0_3 \\ d_2\mathbf{e}_1 + L_2\mathbf{e}_3 \\ 0_3 \end{bmatrix} \in \Omega \qquad \text{or likewise} \qquad \begin{bmatrix} 0_3 \\ 0_3 \\ \mathbf{e}_1 \\ 0_3 \\ \mathbf{e}_3 \\ 0_3 \\ \mathbf{e}_3 \\ 0_3 \end{bmatrix} \in \Omega_{\mathbf{y}} \,.$$

This chart is then used in Chapter 4 when discussing the stability of the system through a linearization process.

3.2 Parametrization of the unit vectors

In our mathematical model, we accept that the thickness of the load is negligible with respect to its length, and a similar simplification can be reasonably done for the cable.

Therefore, rather than a full rotation matrix, their orientation can be also described with a unit vector. In particular, we choose to define $\mathbf{n}_B$, $\mathbf{n}_{C1}$ and $\mathbf{n}_{C2}$ as rotations of a standard unit vector with respect to a fixed reference frame, following the roll-pitch-yaw convention detailed in Appendix :

$$\mathbf{n}_B = \mathbf{R}_z(\psi)\, \mathbf{R}_y(\theta)\, \mathbf{e}_1 = \begin{bmatrix} \cos\theta\cos\psi \\ \cos\theta\sin\psi \\ -\sin\theta \end{bmatrix} \tag{3.6}$$

$$\mathbf{n}_{Ci} = \mathbf{R}_y(\theta_i)\, \mathbf{R}_x(\varphi_i)\, \mathbf{e}_3 = \begin{bmatrix} \cos\varphi_i \sin\theta_i \\ -\sin\varphi_i \\ \cos\varphi_i\cos\theta_i \end{bmatrix} \qquad i = 1, 2\,. \tag{3.7}$$

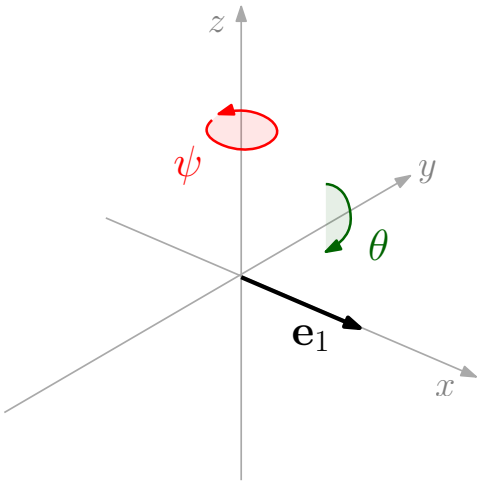


Figure 3.1: Extrinsic rotation of $\mathbf{e}_1$ with respect to ψ, θ

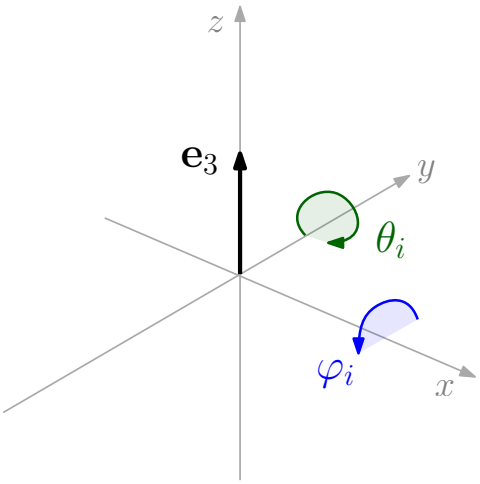


Figure 3.2: Extrinsic rotation of $\mathbf{e}_3$ with respect to φ_i, θ_i

The rotations in (3.6) and (3.7) are represented in Figure 3.1 and Figure 3.2 respectively. Note that with this choice of parametrization, when no rotations occur (null values of all Euler angles), then $\mathbf{n}_B = \mathbf{e}_1$ and $\mathbf{n}_{C1} = \mathbf{n}_{C2} = \mathbf{e}_3$.

Domain of the unit vector functions

Equations (3.6) and (3.7) show that the unit vectors can be written as functions of the Euler angles. In order to define a chart from a region of the manifold Ω to a subset of the Euclidean space $\mathbb{R}^{18}$, the Euler angles must be mapped to the corresponding unit vector through a continuous bijective function. This is achieved by properly choosing the range of Euler angle values:

$$\begin{aligned} n^B \quad : \quad (-\pi,\pi)\times\left(-\frac{\pi}{2},\frac{\pi}{2}\right) \quad &\longrightarrow \quad \mathbb{S}^2_B = \mathbb{S}^2 \backslash \left\{ \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \in \mathbb{S}^2 \,:\, n_1 < 0\,,\; n_2 = 0 \right\} \\ \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad &\longmapsto \quad \begin{bmatrix} \cos\beta\cos\alpha \\ \cos\beta\sin\alpha \\ -\sin\beta \end{bmatrix} \end{aligned} \tag{3.8}$$

$$\begin{aligned} n^C \quad : \quad \left(-\frac{\pi}{2},\frac{\pi}{2}\right)\times(-\pi,\pi) \quad &\longrightarrow \quad \mathbb{S}^2_C = \mathbb{S}^2 \backslash \left\{ \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \in \mathbb{S}^2 \,:\, n_1 = 0\,,\; n_3 < 0 \right\} \\ \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad &\longmapsto \quad \begin{bmatrix} \cos\alpha\sin\beta \\ -\sin\alpha \\ \cos\alpha\cos\beta \end{bmatrix} \end{aligned} \tag{3.9}$$

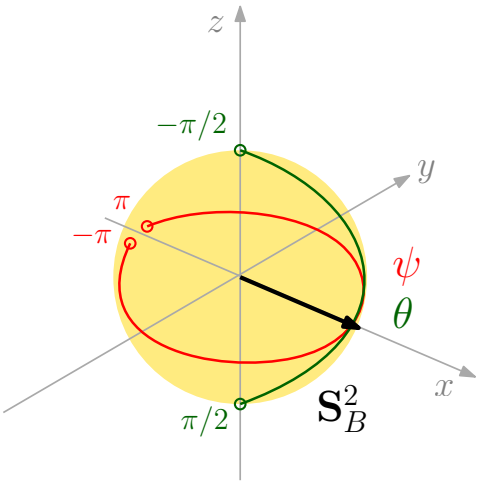


Figure 3.3: Domain and codomain of n^B

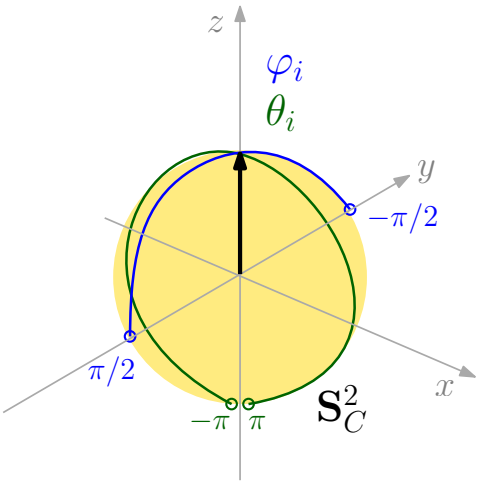


Figure 3.4: Domain and codomain of n^C

where, given the Euler angles (ψ, θ) or (ϕ_i, θ_i) used in the rotations in Figure 3.1 and Figure 3.2, the corresponding unit vectors are unambiguously identified by

$$\mathbf{n}_B = n^B(\psi, \theta) \qquad \text{(3.6 revisited)}$$
$$\mathbf{n}_{Ci} = n^C(\phi_i, \theta_i) \qquad i = 1, 2\,. \qquad \text{(3.7 revisited)}$$

The restrictions on the codomains of (3.8) and (3.9), pictured in Figure 3.3 and Figure 3.4, allow to identify a subset of the state space:

$$\Omega^\xi = \left\{ \mathbf{x} \in \Omega \,:\, \mathbf{n}_B \in \mathbb{S}^2_B \,,\, \mathbf{n}_{Ci} \in \mathbb{S}^2_C \quad i = 1, 2 \right\} . \qquad (3.10)$$

Euler angles

As long as $\mathbf{x} \in \Omega^\xi$, the Euler angles can be expressed as functions of their respective unit vector. This is achieved through the inverse functions of bijections (3.8) and (3.9):

$$\begin{aligned} A^B \;:\; \mathbb{S}^2_B &\longrightarrow (-\pi, \pi) \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \\ \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} &\longmapsto \begin{bmatrix} \operatorname{arctan2}(n_1, n_2) \\ -\arcsin n_3 \end{bmatrix} \end{aligned} \qquad (3.11)$$

$$\begin{aligned} A^C \;:\; \mathbb{S}^2_C &\longrightarrow \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \times (-\pi, \pi) \\ \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} &\longmapsto \begin{bmatrix} -\arcsin n_2 \\ \operatorname{arctan2}(n_3, n_1) \end{bmatrix} \end{aligned} \qquad (3.12)$$

where, given the unit vector, the Euler angle pairs (ψ, θ) or (ϕ_i, θ_i) used in the rotations in Figure 3.1 and Figure 3.2 are unambiguously identified by

$$\begin{bmatrix}\psi \\ \theta\end{bmatrix} = A^B(\mathbf{n}_B) \tag{3.13}$$

$$\begin{bmatrix}\varphi_i \\ \theta_i\end{bmatrix} = A^C(\mathbf{n}_{Ci}) \qquad i = 1, 2\,. \tag{3.14}$$

Time derivatives of the Euler Angles

The time derivatives of of the Euler angles can be expressed as functions of the unit vector and its angular velocity. By deriving (3.11), through the chain rule we obtain

$$\frac{d}{dt}A^B(\mathbf{n}) = \frac{\partial A^B}{\partial \mathbf{n}} \cdot \frac{d\mathbf{n}}{dt} = \frac{\partial A^B}{\partial \mathbf{n}} \cdot (\boldsymbol{\omega} \times \mathbf{n}) \tag{3.15}$$

where $\mathbf{n}$ is a generic unit vector and $\boldsymbol{\omega}$ is its angular velocity. A similar result can be obtained from the time derivative of (3.12).

These computations, together with the simplifications made possible by (2.8a) and (2.8b), yield the explicit expressions

$$\begin{array}{rccc} \Delta^B & : & \mathbb{S}^2_B \times \mathbb{R}^3 & \longrightarrow & \mathbb{R} \times \mathbb{R} \\ & & \left(\begin{bmatrix}n_1 \\ n_2 \\ n_3\end{bmatrix}, \begin{bmatrix}\omega_1 \\ \omega_2 \\ \omega_3\end{bmatrix}\right) & \longmapsto & \begin{bmatrix}\frac{\omega_3}{n_1^2+n_2^2} \\ \frac{n_1\omega_2 - n_2\omega_1}{\sqrt{n_1^2+n_2^2}}\end{bmatrix} \end{array} \tag{3.16}$$

$$\begin{array}{rccc} \Delta^C & : & \mathbb{S}^2_C \times \mathbb{R}^3 & \longrightarrow & \mathbb{R} \times \mathbb{R} \\ & & \left(\begin{bmatrix}n_1 \\ n_2 \\ n_3\end{bmatrix}, \begin{bmatrix}\omega_1 \\ \omega_2 \\ \omega_3\end{bmatrix}\right) & \longmapsto & \begin{bmatrix}\frac{n_3\omega_1 - n_1\omega_3}{\sqrt{n_1^2+n_3^2}} \\ \frac{\omega_2}{n_1^2+n_3^2}\end{bmatrix} \end{array} \tag{3.17}$$

and in particular, if the pairs (ψ, θ) and (ϕ_i, θ_i) are defined according to (3.13) and (3.14), then clearly

$$\begin{bmatrix}\dot{\psi} \\ \dot{\theta}\end{bmatrix} = \Delta^B(\mathbf{n}_B\,, \boldsymbol{\omega}_B) \tag{3.18}$$

$$\begin{bmatrix}\dot{\varphi}_i \\ \dot{\theta}_i\end{bmatrix} = \Delta^C(\mathbf{n}_{Ci}\,, \boldsymbol{\omega}_{Ci}) \qquad i = 1, 2\,. \tag{3.19}$$

3.3 Parametrization of the angular velocities

Let $\mathbf{n}$ designate a generic unit vector with a nonzero angular velocity $\boldsymbol{\omega}$. Then, by definition the time derivative of the unit vector is given by

$$\dot{\mathbf{n}} = \boldsymbol{\omega} \times \mathbf{n}\,.$$

Additionally, if $\mathbf{n}$ and $\boldsymbol{\omega}$ are orthogonal, then the trio $\dot{\mathbf{n}}$, $\boldsymbol{\omega}$ and $\mathbf{n}$ form a basis in $\mathbb{R}^3$ and the following equalities hold:

$$\boldsymbol{\omega} = \mathbf{n} \times \dot{\mathbf{n}}$$

$$\mathbf{n} = \frac{\dot{\mathbf{n}} \times \boldsymbol{\omega}}{\|\dot{\mathbf{n}}\|\,\|\boldsymbol{\omega}\|}$$

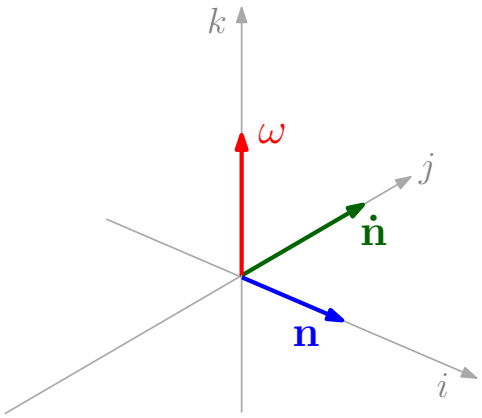


Figure 3.5: Diagram showing the relation between $\dot{\mathbf{n}}$, $\boldsymbol{\omega}$ and $\mathbf{n}$

Therefore, because of the constraint enforced by (2.8b), $\boldsymbol{\omega}_B$ can be expressed in terms of $\mathbf{n}_B$ and $\dot{\mathbf{n}}_B$, which in turn can be obtained as functions of the Euler angles:

$$\begin{aligned}\boldsymbol{\omega}_B &= \mathbf{n}_B \times \dot{\mathbf{n}}_B \\ &= n^B(\psi,\theta) \times \frac{\partial}{\partial t}\, n^B(\psi,\theta) \\ &= n^B(\psi,\theta) \times \frac{\partial n^B}{\partial(\psi,\theta)} \cdot \begin{bmatrix}\dot{\psi}\\ \dot{\theta}\end{bmatrix}\end{aligned} \tag{3.20}$$

Likewise, since (2.8d) is equivalent to $\mathbf{n}_{Ci} \perp \boldsymbol{\omega}_{Ci}$, the angular velocities of the cables can be obtained from

$$\boldsymbol{\omega}_{Ci} = n^C(\varphi_i,\theta_i) \times \frac{\partial n^C}{\partial(\varphi_i,\theta_i)} \cdot \begin{bmatrix}\dot{\varphi}_i\\ \dot{\theta}_i\end{bmatrix} \tag{3.21}$$

The expressions that appear in (3.20) and (3.21) allow us to define the angular velocity functions:

$$\begin{aligned}\omega^B \quad : \quad (-\pi,\pi) \times \left(-\frac{\pi}{2},\frac{\pi}{2}\right) \times \mathbb{R} \times \mathbb{R} \quad &\longrightarrow \quad \mathbb{R}^3 \\ \left(\begin{bmatrix}\alpha\\ \beta\end{bmatrix}, \begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\end{bmatrix}\right) \quad &\longmapsto \quad \begin{bmatrix}\dot{\alpha}\cos\alpha\cos\beta\sin\beta - \dot{\beta}\sin\alpha\\ \dot{\alpha}\sin\alpha\cos\beta\sin\beta + \dot{\beta}\cos\alpha\\ \dot{\alpha}\cos^2\beta\end{bmatrix}\end{aligned} \tag{3.22}$$

$$\begin{aligned}\omega^C \quad : \quad \left(-\frac{\pi}{2},\frac{\pi}{2}\right) \times (-\pi,\pi) \times \mathbb{R} \times \mathbb{R} \quad &\longrightarrow \quad \mathbb{R}^3 \\ \left(\begin{bmatrix}\alpha\\ \beta\end{bmatrix}, \begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\end{bmatrix}\right) \quad &\longmapsto \quad \begin{bmatrix}\dot{\beta}\sin\beta\cos\alpha\sin\alpha + \dot{\alpha}\cos\beta\\ \dot{\beta}\cos^2\alpha\\ \dot{\beta}\cos\beta\cos\alpha\sin\alpha - \dot{\alpha}\sin\beta\end{bmatrix}\end{aligned} \tag{3.23}$$

3.4 Coordinate chart

The various functions delineated in the previous sections establish a one to one correspondence

$$\begin{aligned}(\mathbf{n}_B\,,\,\boldsymbol{\omega}_B) &\longleftrightarrow \big(\psi\,,\,\theta\,,\,\dot\psi\,,\,\dot\theta\big)\\ (\mathbf{n}_{Ci}\,,\,\boldsymbol{\omega}_{Ci}) &\longleftrightarrow \big(\varphi_i\,,\,\theta_i\,,\,\dot\varphi_i\,,\,\dot\theta_i\big) \qquad\qquad i=1,2.\end{aligned}$$

This allows us to define a chart ξ between a region of the manifold,

$$\Omega^\xi = \big\{\mathbf{x}\in\Omega\,:\,\mathbf{n}_B\in\mathbb{S}^2_B\,,\,\mathbf{n}_{Ci}\in\mathbb{S}^2_C\quad i=1,2\big\} \tag{3.10 revisited}$$

and a subset of $\mathbb{R}^{18}$

$$\Omega^\xi_{\mathbf{z}} = \big\{\mathbf{z}\in\mathbb{R}^{18}\,:\,(\psi,\theta)\;\in(-\pi,\pi)\times\Big(-\frac{\pi}{2},\frac{\pi}{2}\Big) \tag{3.24a}$$

$$(\varphi_i,\theta_i)\in\Big(-\frac{\pi}{2},\frac{\pi}{2}\Big)\times(-\pi,\pi)\qquad i=1,2\,\big\} \tag{3.24b}$$

where each element in $\Omega^\xi_{\mathbf{z}}$ has the following structure:

$$\mathbf{z}=\begin{bmatrix}\mathbf{p}\\ \mathbf{v}\\ \psi\\ \theta\\ \dot\psi\\ \dot\theta\\ \varphi_1\\ \theta_1\\ \dot\varphi_1\\ \dot\theta_1\\ \varphi_2\\ \theta_2\\ \dot\varphi_2\\ \dot\theta_2\end{bmatrix}$$

Specifically, the explicit expression of ξ is given by

$$\begin{aligned}\xi\quad:\qquad\qquad & \Omega^\xi \;\longrightarrow\; \Omega^\xi_{\mathbf{z}}\\ \begin{bmatrix}\mathbf{p}\\ \mathbf{v}\\ \mathbf{n}_B\\ \boldsymbol{\omega}_B\\ \mathbf{p}+d_1\,\mathbf{n}_B+L_1\,\mathbf{n}_{C1}\\ \mathbf{v}+d_1\,\boldsymbol{\omega}_B\times\mathbf{n}_B+L_1\,\boldsymbol{\omega}_{C1}\times\mathbf{n}_{C1}\\ \mathbf{p}+d_2\,\mathbf{n}_B+L_2\,\mathbf{n}_{C2}\\ \mathbf{v}+d_2\,\boldsymbol{\omega}_B\times\mathbf{n}_B+L_2\,\boldsymbol{\omega}_{C2}\times\mathbf{n}_{C2}\end{bmatrix} &\;\longmapsto\; \begin{bmatrix}\mathbf{p}\\ \mathbf{v}\\ a^B(\mathbf{n}_B)\\ \Delta^B(\mathbf{n}_B,\boldsymbol{\omega}_B)\\ a^C(\mathbf{n}_{C1})\\ \Delta^C(\mathbf{n}_{C1},\boldsymbol{\omega}_{C1})\\ a^C(\mathbf{n}_{C2})\\ \Delta^C(\mathbf{n}_{C2},\boldsymbol{\omega}_{C2})\end{bmatrix}\end{aligned} \tag{3.25}$$

and as for the inverse function

$$\xi^{-1} \quad : \qquad \Omega^{\xi}_{\mathbf{z}} \quad \longrightarrow \quad \Omega^{\xi}$$

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \psi \\ \theta \\ \dot{\psi} \\ \dot{\theta} \\ \varphi_1 \\ \theta_1 \\ \dot{\varphi}_1 \\ \dot{\theta}_1 \\ \varphi_2 \\ \theta_2 \\ \dot{\varphi}_2 \\ \dot{\theta}_2 \end{bmatrix} \quad \longmapsto \quad \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ n^B(\psi,\theta) \\ \omega^B(\psi,\theta,\dot{\psi},\dot{\theta}) \\ \mathbf{p} + d_1\, n^B(\psi,\theta) + L_1\, n^C(\varphi_1,\theta_1) \\ \mathbf{v} + d_1\, \omega^B(\psi,\theta,\dot{\psi},\dot{\theta}) \times n^B(\psi,\theta) + L_1\, \omega^C(\varphi_1,\theta_1,\dot{\varphi}_1,\dot{\theta}_1) \times n^C(\varphi_1,\theta_1) \\ \mathbf{p} + d_2\, n^B(\psi,\theta) + L_2\, n^C(\varphi_2,\theta_2) \\ \mathbf{v} + d_2\, \omega^B(\psi,\theta,\dot{\psi},\dot{\theta}) \times n^B(\psi,\theta) + L_2\, \omega^C(\varphi_2,\theta_2,\dot{\varphi}_2,\dot{\theta}_2) \times n^C(\varphi_2,\theta_2) \end{bmatrix}$$

Chapter 4

Equilibria and stability

In this chapter we identify a subset of equilibrium points and compute the corresponding UAV thrusts. Afterwards, we carry out a preliminary study on the stability of the system in one of the accepted equilibrium configurations.

4.1 Equilibrium configurations

Given the vector field as it is defined in (2.26),

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

equilibrium is achieved for pairs in the set

$$\left\{ (\mathbf{x}, \mathbf{u}) \in \Omega \times \left(\mathbb{R}^3\right)^2 \,:\, \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = 0 \right\} . \tag{4.1}$$

The first obvious observation is that the velocities associated to any and all equilibrium states must be zero:

$$\dot{\mathbf{x}} = 0 \Longrightarrow \begin{cases} \dot{\mathbf{p}} = \mathbf{v} = 0 \\ \dot{\mathbf{n}}_B = \boldsymbol{\omega}_B \times \mathbf{n}_B = 0 \\ \dot{\mathbf{p}}_1 = \mathbf{v}_1 = 0 \\ \dot{\mathbf{p}}_2 = \mathbf{v}_2 = 0 \end{cases}$$

where the second condition, together with (2.8a) and (2.8b), implies $\boldsymbol{\omega}_B = 0$.

Even taking into account all the simplifications that come from having null velocities, finding all the pairs in (4.1) is not trivial. On the other hand, this is not necessarily part of the goal of this thesis work, as the stabilization can be achieved in a subset of (4.1).

In fact, in a realistic scenario, the only acceptable thrusts $\mathbf{u}_1$ and $\mathbf{u}_2$ are those associated with zero roll and pitch angles for both UAVs. Clearly, this only happens when both $\mathbf{u}_1$ and $\mathbf{u}_2$ are aligned with $\mathbf{e}_3$. Therefore, for the equilibrium input, we focus on the subset

$$\mathbf{u}_i \in \operatorname{span}\{\mathbf{e}_3\} \qquad i = 1, 2 \,. \tag{4.2}$$

This restriction on the thrusts gives a lot of insight on the state. Note in fact that any and all equilibrium states associated to (4.2) must also satisfy

$$\mathbf{n}_{Ci} = \mathbf{e}_3 \qquad i = 1, 2 \,. \tag{4.3}$$

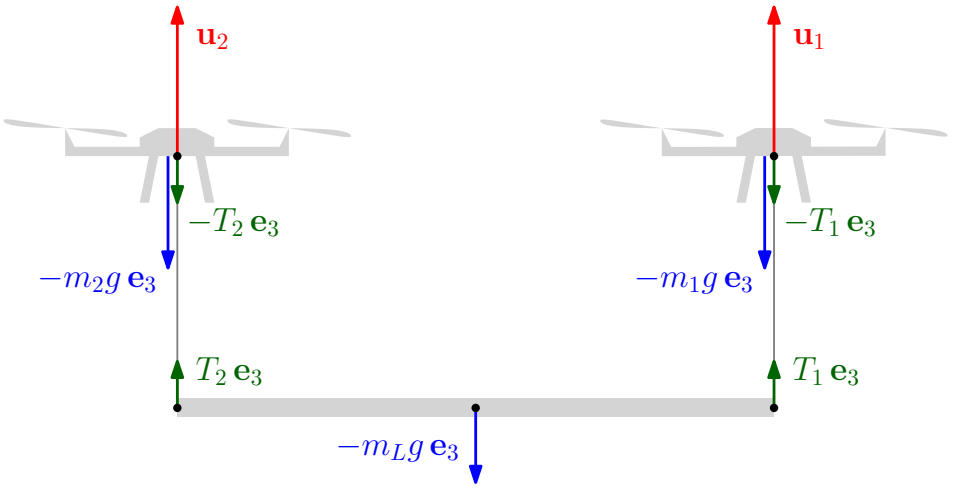


Figure 4.1: Representation of the forces acting on the system when (4.2) holds and $\mathbf{x} \in \Gamma$

Indeed, if (4.2) holds but (4.3) does not, then the tensions have a non compensated horizontal component which pulls the UAVs, meaning that they have nonzero linear velocities $\mathbf{v}_1$ and $\mathbf{v}_2$.

With all this information about the equilibrium states associated with vertical thrusts, we define the subset of the state space

$$\begin{aligned}
\Gamma = \big\{ \mathbf{x} \in \Omega \,:\, & \mathbf{v} = 0 && && \text{(4.4a)}\\
& \mathbf{n}_B \neq \pm \mathbf{e}_3 && && \text{(4.4b)}\\
& \boldsymbol{\omega}_B = 0 && && \text{(4.4c)}\\
& \mathbf{p}_i - \mathbf{p}_{Bi} = L_i\,\mathbf{e}_3 && i = 1,2 && \text{(4.4d)}\\
& \mathbf{v}_i = 0 && i = 1,2 \big\}\,. && \text{(4.4e)}
\end{aligned}$$

where evidently (4.4d) is equivalent to (4.3).

We now compute the magnitude of the thrust vectors (4.2) associated to any and all $\mathbf{x} \in \Gamma$, and show that it is a constant that only depends on the geometric characteristics of the system.

As can be seen in Figure 4.1, if (4.2) and (4.3) hold then all the relevant forces (i.e. tensions along the cables, UAV upward thrusts and gravity) are aligned with $\mathbf{e}_3$. As a consequence of this choice, the system of Newton equations (2.13) takes the much simpler form

$$\begin{cases}
\mathbf{u}_1 = (T_1 - m_1 g)\,\mathbf{e}_3 & \text{(4.5a)}\\
\mathbf{u}_2 = (T_2 - m_2 g)\,\mathbf{e}_3 & \text{(4.5b)}\\
m_L g\,\mathbf{e}_3 = (T_1 + T_2)\,\mathbf{e}_3 & \text{(4.5c)}\\
0 = (d_1 T_1 + d_2 T_2)\,\mathbf{n}_B \times \mathbf{e}_3 & \text{(4.5d)}
\end{cases}$$

Equation (4.5d), together with condition (4.4b), allows to characterize the relation between the tensions as

$$T_1 = \frac{-d_2}{d_1} T_2 \tag{4.6}$$

and combining this result with (4.5c) we obtain

$$T_1 = \frac{d_2}{d_2 - d_1} m_L g \tag{4.7}$$

$$T_2 = \frac{d_1}{d_1 - d_2} m_L g \,. \tag{4.8}$$

By plugging the values of the tensions (4.7) and (4.8) in the equations (4.5a) and (4.5b) respectively, we finally obtain:

$$\mathbf{u_{eq},1} = \left(m_1 + \frac{d_2}{d_2 - d_1} m_L \right) g \, \mathbf{e}_3 \tag{4.9}$$

$$\mathbf{u_{eq},2} = \left(m_2 + \frac{d_1}{d_1 - d_2} m_L \right) g \, \mathbf{e}_3 \,. \tag{4.10}$$

Notice how, as stated previously, when $\mathbf{x} \in \Gamma$ and (4.2), both of the thrusts have a constant magnitude that is independent of the state.

The entire set Γ is made up of equilibrium points. Let the equilibrium input be defined as the stacked vector

$$\mathbf{u_{eq}} = \begin{bmatrix} \mathbf{u_{eq},1} \\ \mathbf{u_{eq},2} \end{bmatrix} \tag{4.11}$$

then it can be proven that

$$\mathbf{f}\left(\mathbf{x}, \mathbf{u_{eq}}\right) = 0 \quad \forall \, \mathbf{x} \in \Gamma \,.$$

This can be easily verified with the aid of softwares such as Wolfram Mathematica or MATLAB with its symbolic toolbox.

4.2 Stability in the equilibrium set

In this thesis work, the study of stability is divided in two stages. First, we conduct a preliminary analysis and prove that the mathematical model can be stabilized with a basic PD controller. Then, in Chapter 6, we switch to a more realistic model, simulated in Gazebo, in order to develop and test a controller with improved stabilization capabilities.

As far as the preliminary analysis is concerned, we discuss the stability via linearization of the closed loop vector field in the point

$$\mathbf{x_{eq}} = \begin{bmatrix} 0_3 \\ 0_3 \\ \mathbf{e}_1 \\ 0_3 \\ d_1 \mathbf{e}_1 + L_1 \mathbf{e}_3 \\ 0_3 \\ d_2 \mathbf{e}_1 + L_2 \mathbf{e}_3 \\ 0_3 \end{bmatrix} \in \Gamma \,. \tag{4.12}$$

Note that from studying the behavior of the system in $\mathbf{x_{eq}}$ we are able to extend the results to any point in the subset

$$\Gamma^{\parallel} = \left\{ \mathbf{x} \in \Gamma \, : \, \mathbf{n}_B^{\intercal} \mathbf{e}_3 = 0 \right\} , \tag{4.13}$$

which is comprised of the equilibrium configurations in Γ associated to a horizontal orientation of the bar.

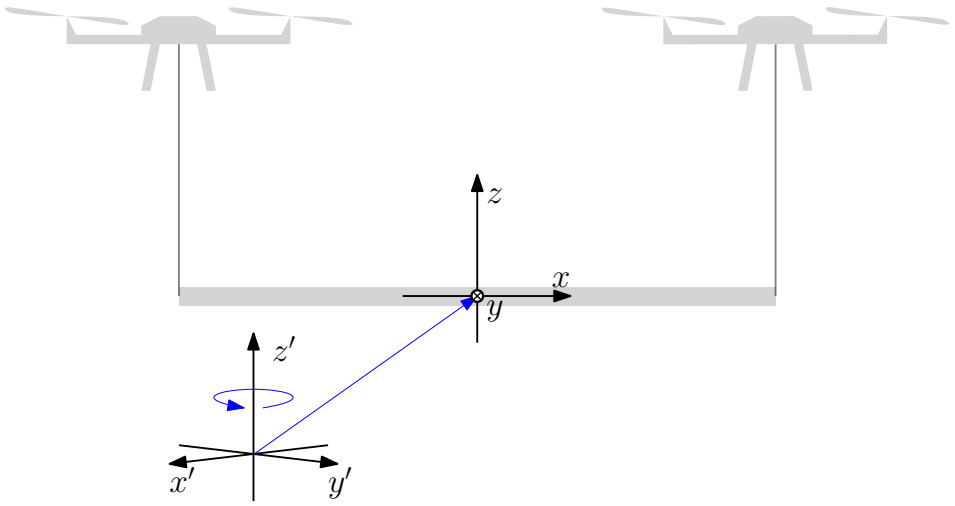


Figure 4.2: Frame xyz sees the system in $\mathbf{x_{eq}}$, while $x'y'z'$ sees it in a generic $\mathbf{x} \in \Gamma^{\parallel}$

Indeed, the set $\Gamma^{\parallel}$ is defined in such a way that all its points can be traced back to the same physical configuration of the UAVs-load system. Let us consider the Euler angles parametrization of the equilibrium points,

$$\xi(\mathbf{x}) = \begin{bmatrix} \mathbf{p} \\ 0_3 \\ \psi \\ \theta \\ 0_{10} \end{bmatrix} \forall \mathbf{x} \in \Gamma \qquad \text{and in particular} \qquad \xi(\mathbf{x}) = \begin{bmatrix} \mathbf{p} \\ 0_3 \\ \psi \\ 0 \\ 0_{10} \end{bmatrix} \forall \mathbf{x} \in \Gamma^{\parallel}\,. \tag{4.14}$$

As long as $\mathbf{x} \in \Gamma^{\parallel}$, the UAVs-load system can be seen as a rigid body, whose internal structure is not altered by translations (changes to $\mathbf{p}$) nor rotations about the vertical axis (changes to ψ). The concept that the system remains the same is also shown in Figure 4.2, where the difference between $\mathbf{x_{eq}}$ and a generic point $\mathbf{x} \in \Gamma^{\parallel}$ is traced back to a translation and yaw rotation of the external reference frame.

This correlation between the points in $\Gamma^{\parallel}$ implies that, if the linearized system is asymptotically stable at $\mathbf{x_{eq}}$, then that same control law can be used to stabilize the system around any reference configuration $\mathbf{x}^{\mathrm{REF}} \in \Gamma^{\parallel}$.

4.3 Closed loop vector field

Let the vehicle position error and its derivative be respectively defined as

$$\begin{aligned} \mathbf{e}_{\mathbf{p}_i} &= \mathbf{p}_{\mathbf{eq},i} - \mathbf{p}_i && i = 1, 2 \\ \dot{\mathbf{e}}_{\mathbf{p}_i} &= \mathbf{v}_{\mathbf{eq},i} - \mathbf{v}_i && i = 1, 2 \end{aligned}$$

where the equilibrium values are those in (4.12):

$$\begin{aligned} \mathbf{p}_{\mathbf{eq},i} &= d_i \mathbf{e}_1 + L_i \mathbf{e}_3 && i = 1, 2 \\ \mathbf{v}_{\mathbf{eq},i} &= 0 && i = 1, 2\,. \end{aligned}$$

In order to study the stability of the system, we introduce a simple control law, which is then improved in Chapter 6. Consider the basic PD controller

$$\mathbf{u}_{\mathrm{PD}}(\mathbf{e}_{\mathbf{p}_i}) = \mathrm{K}_P\, \mathbf{e}_{\mathbf{p}_i} + \mathrm{K}_D\, \dot{\mathbf{e}}_{\mathbf{p}_i} \tag{4.15}$$

where K_P and K_D are diagonal matrices in order to allow different gains for the three axes:

$$\mathrm{K}_P = \begin{bmatrix} k_{Px} & 0 & 0 \\ 0 & k_{Py} & 0 \\ 0 & 0 & k_{Pz} \end{bmatrix} \qquad \mathrm{K}_D = \begin{bmatrix} k_{Dx} & 0 & 0 \\ 0 & k_{Dy} & 0 \\ 0 & 0 & k_{Dz} \end{bmatrix}.$$

The control input is given by

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} \mathbf{u}_{\mathrm{PD}}(\mathbf{e}_{\mathbf{p}_1}) \\ \mathbf{u}_{\mathrm{PD}}(\mathbf{e}_{\mathbf{p}_2}) \end{bmatrix} + \mathbf{u}_{\mathbf{eq}} \tag{4.16}$$

where the PD term is used to zero the position error, and $\mathbf{u}_{\mathbf{eq}}$ contains the thrusts necessary to lift the total mass of the UAVs and the load.

Once the control input can be written as a function of the state, the closed loop vector field is defined as

$$\dot{\mathbf{x}} = \mathbf{f}^{\mathrm{CL}}(\mathbf{x}) = \mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x}))\,. \tag{4.17}$$

4.4 Linearization

If we linearize the vector field (4.17) about the equilibrium point $\mathbf{x}_{\mathbf{eq}}$, the resulting system falls in the uncertain case of the linearization theorem.

The reason lies with the fact that the representation of the system in the state space Ω contains some degree of redundancy. As we discussed in Chapter 3, by using an atlas every state $\mathbf{x} \in \Omega$ can be expressed in terms of the Euler angles, going from a 24-dimensional representation to a more concise 18-dimensional one.

Other than mapping the region Ω^ξ of manifold Ω in a subset $\Omega^\xi_{\mathbf{z}}$ of the Euclidean space $\mathbb{R}^{18}$, the chart ξ also allows to study the system dynamics in the neighborhood of $\mathbf{x}_{\mathbf{eq}}$:

$$\dot{\mathbf{z}} = \frac{d}{dt}\xi(\mathbf{x}) = \frac{\partial \xi}{\partial \mathbf{x}} \cdot\ \mathbf{f}(\mathbf{x}, \mathbf{u}) \coloneqq \widetilde{\mathbf{f}}_{\mathbf{z}}(\mathbf{x}, \mathbf{u}) \tag{4.18}$$

$$= \widetilde{\mathbf{f}}_{\mathbf{z}}(\xi^{-1}(\mathbf{z}), \mathbf{u}) \coloneqq \mathbf{f}_{\mathbf{z}}(\mathbf{z}, \mathbf{u}) \tag{4.19}$$

and specifically the closed loop vector field corresponding to the PD controller (4.16) is described by

$$\dot{\mathbf{z}} = \mathbf{f}^{\mathrm{CL}}_{\mathbf{z}}(\mathbf{z}) = \widetilde{\mathbf{f}}_{\mathbf{z}}(\mathbf{x}, \mathbf{u}(\mathbf{x}))\Big|_{\mathbf{x}=\xi^{-1}(\mathbf{z})}\,. \tag{4.20}$$

Indeed, if we switch to $\mathbf{z} \in \Omega^\xi_{\mathbf{z}}$, the linearization theorem can be used to prove asymptotic stability. The dynamics of the system described by the closed loop vector field (4.20) about the equilibrium point

$$\mathbf{z}_{\mathbf{eq}} = \xi(\mathbf{x}_{\mathbf{eq}}) = 0_{18} \tag{4.21}$$

are approximated by the linear system

$$\dot{\mathbf{z}} \ \backsimeq\ \mathbf{F}^{\mathrm{CL}}_{\mathbf{z}}\, \mathbf{z} \tag{4.22}$$

where $\mathbf{F}^{\mathrm{CL}}_{\mathbf{z}}$ is the Jacobian matrix of $\mathbf{f}^{\mathrm{CL}}_{\mathbf{z}}(\mathbf{z})$ evaluated in $\mathbf{z}_{\mathbf{eq}}$:

$$\mathbf{F}^{\mathrm{CL}}_{\mathbf{z}} = \left[\frac{\partial}{\partial \mathbf{z}}\, \mathbf{f}^{\mathrm{CL}}_{\mathbf{z}}(\mathbf{z})\right]_{\mathbf{z}=0_{18}}. \tag{4.23}$$

Change of basis

We now discuss the stability of the linearized system in the particular case where

$$m_1 = m_2 \tag{4.24a}$$
$$d_1 = -d_2 \tag{4.24b}$$
$$L_1 = L_2\,. \tag{4.24c}$$

These additional assumptions are not especially difficult to approach in a realistic scenario, and are especially advantageous for simplifying the structure of matrix $\mathbf{F}_{\mathbf{z}}^{\mathrm{CL}}$, since they allow to decouple the dynamic equations of the linear system.

Specifically, under these conditions there exists a change of basis through which the dynamics can be expressed through chains of integrators. Let the row vector

$$\mathrm{e}_k = \big[\underbrace{0 \quad \cdots \quad 0 \quad 1}_{k} \quad \underbrace{0 \quad \cdots \quad 0}_{18-k}\big]$$

designate the transpose of a standard unit vector in $\mathbb{R}^{18}$. Then, the change of basis matrix is defined as

$$\mathbf{P} = \begin{bmatrix} \mathrm{e}_3 \\ \mathrm{e}_3 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_8 \\ \mathrm{e}_8 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_1 \\ \mathrm{e}_1 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_1 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_1 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_{12} - \mathrm{e}_{16} \\ (\mathrm{e}_{12} - \mathrm{e}_{16}) \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_2 \\ \mathrm{e}_2 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_2 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_2 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_7 \\ \mathrm{e}_7 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_7 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \\ \mathrm{e}_7 \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \end{bmatrix} \quad \text{such that} \quad \mathbf{P} \cdot \mathbf{z} = \begin{bmatrix} p_z \\ \dot{p}_z \\ \theta \\ \dot{\theta} \\ p_x \\ \dot{p}_x \\ \ddot{p}_x \\ \dddot{p}_x \\ \theta_1 - \theta_2 \\ \dot{\theta}_1 - \dot{\theta}_2 \\ p_y \\ \dot{p}_y \\ \ddot{p}_y \\ \dddot{p}_y \\ \psi \\ \dot{\psi} \\ \ddot{\psi} \\ \dddot{\psi} \end{bmatrix} \quad \text{with} \quad \mathbf{z} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \psi \\ \theta \\ \dot{\psi} \\ \dot{\theta} \\ \varphi_1 \\ \theta_1 \\ \dot{\varphi}_1 \\ \dot{\theta}_1 \\ \varphi_2 \\ \theta_2 \\ \dot{\varphi}_2 \\ \dot{\theta}_2 \end{bmatrix}.$$

In the new basis, the dynamics of the linear system are described by the block diagonal matrix

$$\mathbf{P} \cdot \mathbf{F}_{\mathbf{z}}^{\mathrm{CL}} \cdot \mathbf{P}^{-1} = \begin{bmatrix} \mathbf{F}_{p_z}^{\mathrm{CL}} & & & & & & \\ & \mathbf{F}_{\theta}^{\mathrm{CL}} & & & & & \\ & & \mathbf{F}_{p_x}^{\mathrm{CL}} & & & & \\ & & & \mathbf{F}_{\theta_1-\theta_2}^{\mathrm{CL}} & & & \\ & & & & \mathbf{F}_{p_y}^{\mathrm{CL}} & & \\ & & & & & \mathbf{F}_{\psi}^{\mathrm{CL}} \end{bmatrix} \tag{4.25}$$

where each block is in phase variable form.

Then, as previously indicated, the dynamics of p_z, θ, p_x, $\theta_1 - \theta_2$, p_y and ψ are all decoupled, and are each described by the corresponding submatrix:

$$\mathbf{F}_{p_z}^{\mathrm{CL}} = \begin{bmatrix} 0 & 1 \\ -\frac{2k_{Pz}}{2m_1+m_L} & -\frac{2k_{Dz}}{2m_1+m_L} \end{bmatrix} \tag{4.26}$$

$$\mathbf{F}_{\theta}^{\mathrm{CL}} = \begin{bmatrix} 0 & 1 \\ -\frac{2d_1^2k_{Pz}}{2m_1d_1^2+I_L} & -\frac{2d_1^2k_{Dz}}{2m_1d_1^2+I_L} \end{bmatrix} \tag{4.27}$$

$$\mathbf{F}_{p_x}^{\mathrm{CL}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{gk_{Px}}{L_1m_1} & -\frac{gk_{Dx}}{L_1m_1} & -\frac{2k_{Px}L_1+2gm_1+gm_L}{2L_1m_1} & -\frac{k_{Dx}}{m_1} \end{bmatrix} \tag{4.28}$$

$$\mathbf{F}_{\theta_1-\theta_2}^{\mathrm{CL}} = \begin{bmatrix} 0 & 1 \\ -\frac{2k_{Px}L_1+gm_L}{2L_1m_1} & -\frac{k_{Dx}}{m_1} \end{bmatrix} \tag{4.29}$$

$$\mathbf{F}_{p_y}^{\mathrm{CL}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{gk_{Py}}{L_1m_1} & -\frac{gk_{Dy}}{L_1m_1} & -\frac{2k_{Py}L_1+2gm_1+gm_L}{2L_1m_1} & -\frac{k_{Dy}}{m_1} \end{bmatrix} \tag{4.30}$$

$$\mathbf{F}_{\psi}^{\mathrm{CL}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{d_1^2gk_{Py}m_L}{I_LL_1m_1} & -\frac{d_1^2gk_{Dy}m_L}{I_LL_1m_1} & -\frac{2I_Lk_{Py}L_1+g\left(2m_1d_1^2+I_L\right)m_L}{2I_LL_1m_1} & -\frac{k_{Dy}}{m_1} \end{bmatrix} \tag{4.31}$$

It can be proven, trough the Routh-Hurwitz criterion, that there exist positive values of the gains that make the linearized system is asymptotically stable. Then, in light of the observations made in Section §4.2, we conclude that the controller (4.15) can stabilize the UAVs-load system at any and all configurations included in (4.13) .

Chapter 5

The Gazebo simulator

In this chapter, after introducing the simulation environment, we describe the Gazebo model of the UAVs-load system. Afterwards, we discuss a method to induce disturbances in the simulation, in order to test the control.

5.1 The simulation environment

The choice of the simulation tools used in this thesis work is highly dependent on the need to develop the controller within the already existing framework used in the Smart Mobility Lab at KTH Royal Institute of Technology.

In particular, the existing library of Python controllers relied on the Robot Operating System (ROS) to handle the communication between the workstation and the vehicle. As for the specific version, all the implementation in this thesis work is based on ROS Indigo Igloo, rather than the more recent ROS Kinetic Kame.

A leading factor in choosing ROS Indigo lies with its compatibility with the RotorS package [7] developed by the Autonomous Systems Lab at ETH Zurich, which provides a number of Gazebo models of multirotor vechicles. Specifically, the model of the AscTec Firefly was chosen for both the aerial vehicles required for our load lifting mission.

The simulations are run in the Gazebo 2.2.6 environment, as it is the latest version available that is compatible with ROS Indigo.

5.2 The Universal Robotic Description Format in Gazebo

The Universal Robotic Description Format (URDF) is an XML file format used to describe the elements of a robot in ROS and, by extension, in Gazebo.

The UAVs described in the RotorS version used in the simulations are defined according to the URDF package specifications. For this reason, the ROS implementation created during this thesis work also follows this standard: i.e., the UAVs-load system was modeled through a series of Gazebo compatible URDF files.

In general, the description of a robot in Gazebo is characterized by a series of links connected by joints. As far as the links are concerned, a generic `<link>` element is defined according to the specifics at [8]; in particular, to work in Gazebo a `<link>` must include an `<inertial>` element, with a nonzero `<mass>` value and a (preferably) non singular `<inertia>` matrix. The `<visual>` and `<collision>` elements are not compulsory, and respectively define the visual appearance of the link and its collision properties.

It is very important to point out that the main reference frame of the link coincides with the the reference frame of the associated joint. The `<inertial>`, `<visual>` and `<collision>` elements can all be defined with respect to a different reference frame, whose `<origin>` is expressed in the reference frame of the link. In other words, up to four different reference frames can be associated to each link, as can be see by the example in Figure 5.1.

This is especially useful when defining the inertial properties of the link: generally the origin of the link frame is not in the center of mass of the link, while instead it is desirable to align the inertial frame with the center of mass.

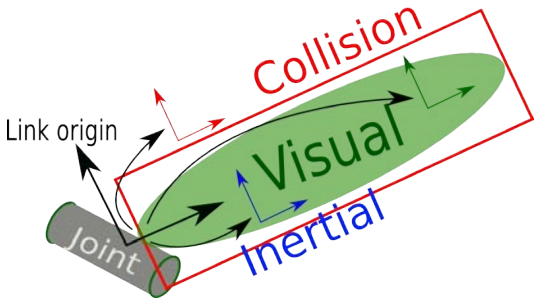


Figure 5.1: Diagram of the different frames associated to a `<link>`

A `<joint>` element is defined according to the specifications at [9]. Essentially, a joint in a URDF file defines a relation between a parent link and a child link, depending on the type of said joint. Each joint is endowed with a reference frame, whose pose is expressed in the reference frame of the parent link, while the child reference frame coincides with the joint reference frame (as previously stated). The relation among these reference frames is described in Figure 5.2.

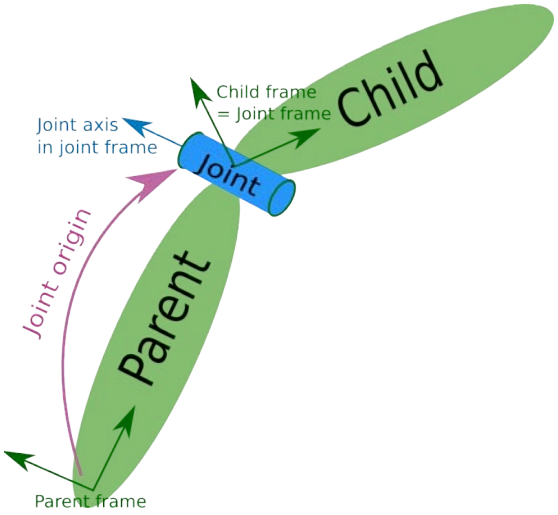


Figure 5.2: Diagram of the joint reference frame, together with the parent and child link reference frames

5.3 URDF model of the system

At its most high level description, the model consists in a link representing the load, joined to two links representing the cables, which are in turn joined to the respective UAV from the RotorS package.

Clearly, for each of the cables, both the joint connected to the vehicle and the joint connected to the load need to allow free movement in two planes at the same time, in order to have the same degree of freedom as a rope in reality. Unfortunately, the URDF format does not support ball joints: therefore, the desired behavior is achieved by overlapping two continuous joints.

Note that the only way to superimpose the joints is with an auxiliary link between them: it is in fact impossible to have two links connected by more than one joint at the same time, because the robot must have a tree structure, in which the links are the nodes and the joints are the edges. The tree associated of the UAVs-load system is featured in Figure 5.3.

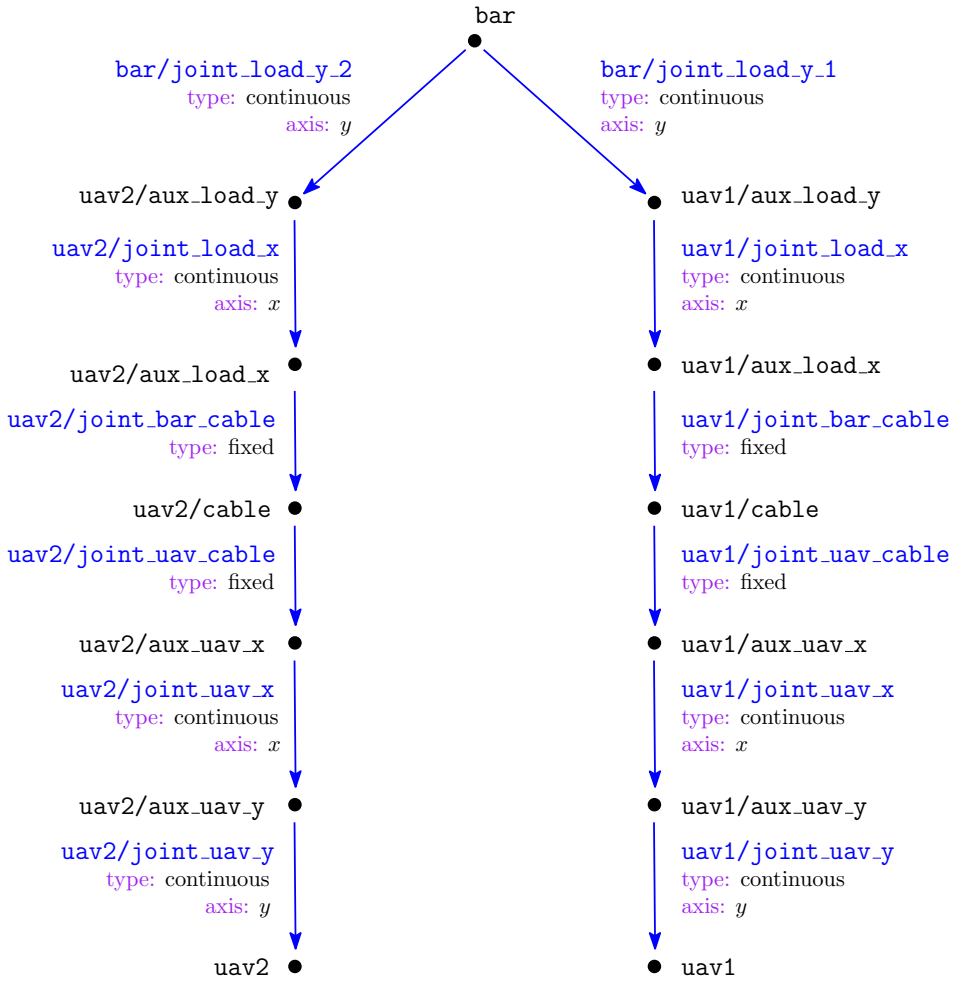


Figure 5.3: Tree structure of the links (nodes) and joints (edges) in the URDF model

Structure of the model

In our URDF model, the vehicles (defined in the RotorS package) are connected to the load by the scripts in the file `cables_and_load.xacro`, which delineates the tree structure of the system.

In order to analyze the graph in Figure 5.3, we go over the contents of `cables_and_load.xacro`, from the root of the tree (the bar link) to the outermost nodes (the vehicles), with a particular emphasis on the joints and the workaround employed to duplicate the behaviour of ball joints.

Starting from the root, two joints connect the fist set of auxiliary sphere links to the two ends of the bar, all the while implementing rotations around the y axis:

```
157 <!-- y axis rotation between cable 1 and load -->
158 <joint name="bar/joint_load_y_1" type="continuous">
159   <origin xyz="${x_load/2 - sphere_radius} 0 ${z_load/2}" rpy="0 0 0"/>
160   <axis xyz="0 1 0"/>
161   <parent link="bar"/>
162   <child link="${link1}"/>
163 </joint>
164
165 <!-- y axis rotation between cable 2 and load -->
166 <joint name="bar/joint_load_y_2" type="continuous">
167   <origin xyz="${-x_load/2 + sphere_radius} 0 ${z_load/2}" rpy="0 0 0" />
168   <axis xyz="0 1 0" />
169   <parent link="bar" />
170   <child link="${link2}" />
171 </joint>
```

From here onwards, the same code is used for both sides. After defining a rotation around the y axis, we define a rotation around x:

```
137 <!-- x axis rotation between cable and load -->
138 <joint name="${robot_namespace}/joint_load_x" type="continuous">
139   <origin xyz="0 0 0" rpy="0 0 0" />
140   <axis xyz="1 0 0" />
141   <parent link="${robot_namespace}/aux_load_y" />
142   <child link="${robot_namespace}/aux_load_x" />
143 </joint>
```

Recall that the joint frame and the child frame always coincide. Therefore at this point there is a continuous joint that allows rotation about the y axis of `aux_load_y`, and another continuous joint that allows rotation about the x axis of `aux_load_x`.

Moreover, since the origin of `joint_load_x` is associated to null values, parent and child link share a reference frame, which in turn case means that `aux_load_y` and `aux_load_x` are overlapped. This stratagem allows two continuous joints about two perpendicular axes to superimpose, allowing rotations around x and y at the same time.

Next, `aux_load_x` is connected to the lower extremity of the `cable` link by a fixed joint. Similarly, the upper extremity of `cable` is attached to `aux_uav_x`, again with a fixed joint:

```
130 <!-- Fixed joint between cable and load -->
131 <joint name="${robot_namespace}/joint_bar_cable" type="fixed">
132   <origin xyz="0 0 ${cable_length/2}" rpy="0 0 0" />
133   <parent link="${robot_namespace}/aux_load_x" />
134   <child link="${robot_namespace}/cable" />
135 </joint>
```

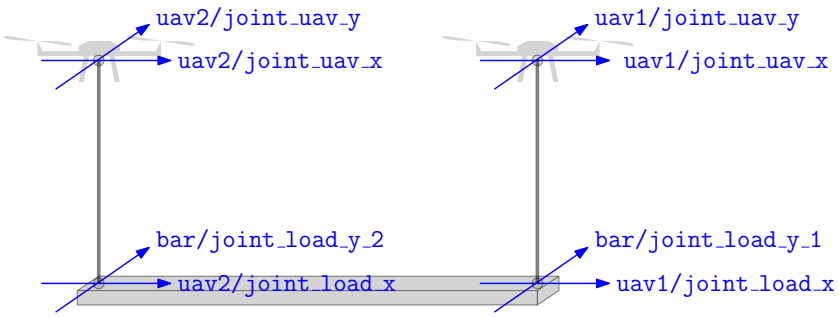


Figure 5.4: Diagram showing the double continuous joints in the URDF model

```
123 <!-- Fixed joint between cable and UAV -->
124 <joint name="${robot_namespace}/joint_uav_cable" type="fixed">
125   <origin xyz="0 0 ${cable_length/2}" rpy="0 0 0" />
126   <parent link="${robot_namespace}/cable" />
127   <child link="${robot_namespace}/aux_uav_x" />
128 </joint>
```

Finally, the connection between `aux_uav_x` and `aux_uav_y` is defined by a continuous joint that allows rotation around the x axis, while the connection between `aux_uav_y` and the body frame of the aerial vehicle is handled by a continuous joint that allows rotation around the y axis. Again, since `aux_uav_x` and `aux_uav_y` occupy the same space, the two joints are superimposed.

```
108 <!-- x axis rotation between UAV and cable -->
109 <joint name="${robot_namespace}/joint_uav_x" type="continuous">
110   <origin xyz="0 0 0" rpy="0 0 0" />
111   <axis xyz="1 0 0" />
112   <parent link="${robot_namespace}/aux_uav_x" />
113   <child link="${robot_namespace}/aux_uav_y" />
114 </joint>
```

```
100   <!-- y axis rotation between UAV and cable -->
101 <joint name="${robot_namespace}/joint_uav_y" type="continuous">
102   <origin xyz="0 0 0" rpy="0 0 0" />
103   <axis xyz="0 1 0" />
104   <parent link="${robot_namespace}/aux_uav_y" />
105   <child link="${uav_body_frame}" />
106 </joint>
```

The global structure of the URDF model is represented in Figure 5.4, showing the geometric shape of the links and the rotation axes of the joints. Note that the simulations have been run with parameters that conform to the simplifications at (4.24).

5.4 Translations and rotations in the Gazebo environment

When the model is spawned in Gazebo, a local reference frame is associated to the root of the robotic tree, which in our case is the load. This reference frame, which we name $\mathcal{XYZ}$, is represented in Figure 5.5, and is such that $\mathcal{X}$ is always aligned with $\mathbf{n}_B$.

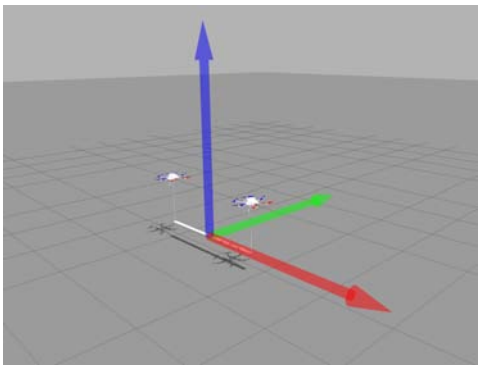


Figure 5.5: Local reference frame in the Gazebo scene

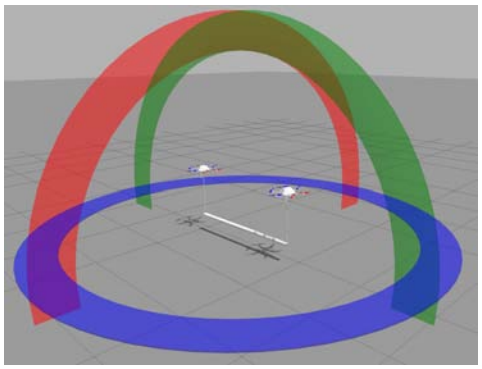


Figure 5.6: Rotations in the Gazebo scene

Among its many functionalities, the simulator allows to manipulate the robot as a whole, through translations and rotations. The latter are pictured in Figure 5.6: in particular, we chose to use Φ, Θ and Ψ to name rotations about $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ respectively.

It is worth pointing out that the states in

$$\Gamma^{\parallel} = \{\mathbf{x} \in \Gamma \,:\, \mathbf{n}_B^{\intercal}\mathbf{e}_3 = 0\}\,, \qquad \text{(4.13 revisited)}$$

can all be described in terms of translations along the axes $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$, together with a Ψ rotation.

5.5 Characterization of the swinging motion of the load

When directing the vehicles towards any target configuration, it is immediately apparent that moving the UAVs-load system almost always implies some kind of swinging motion by the slung load. Therefore, the reliability of the controller is tied with its capability of counterbalancing this behavior.

Thus, it is appropriate to categorize the disturbances that can affect the bar. This is made possible by the concept that any swinging dynamic of the load can be traced back to a combination of three basic oscillatory motions, which in the interest of clarity we name $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$, and which are represented in Figures 5.7, 5.8 and 5.9 respectively.

Note that this particular subdivision is prompted by simulation results, as each of these oscillations is the consequence of what could be considered an elementary motion of the system. In particular,

$\mathcal{S}_1$ is observed with translations along $\mathcal{Y}$ (or with non-destabilizing Φ rotations),

$\mathcal{S}_2$ is observed with translations along $\mathcal{X}$ (or with non-destabilizing Θ rotations),

$\mathcal{S}_3$ is observed with Ψ rotations.

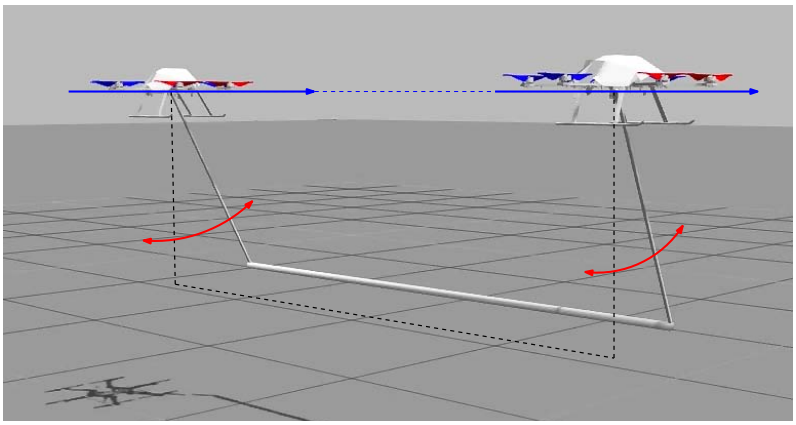


Figure 5.7: Schematic representation of $\mathcal{S}_1$, from `test_1.launch`

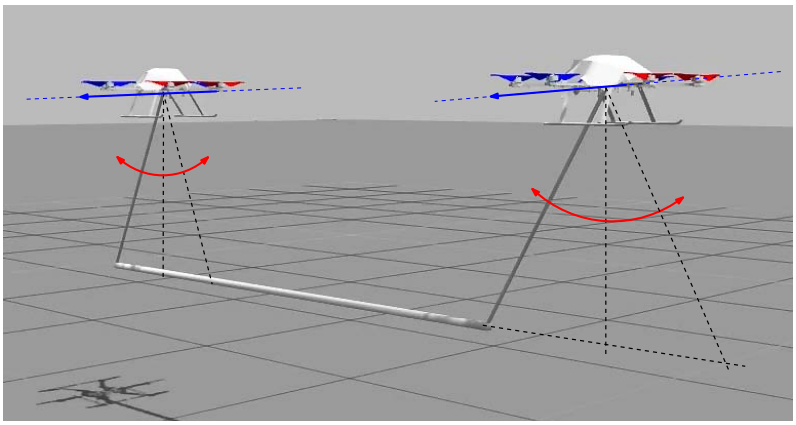


Figure 5.8: Schematic representation of $\mathcal{S}_2$, from `test_2.launch`

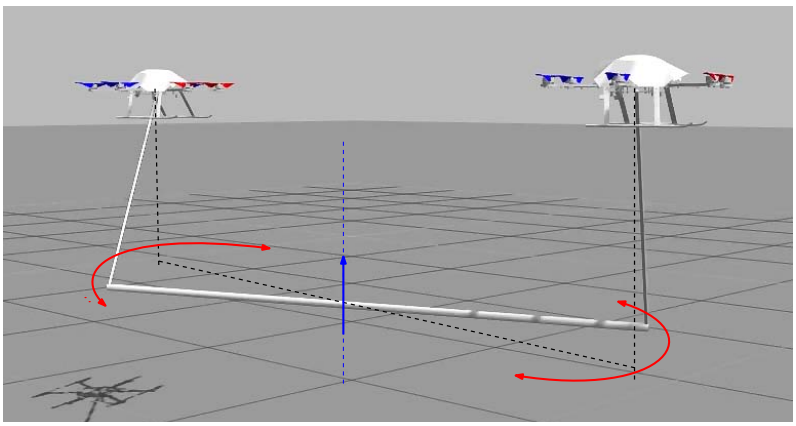


Figure 5.9: Schematic representation of $\mathcal{S}_3$, from `test_3.launch`

URDF model	Launchfiles
uav1/joint_uav_x	j_uav1_x
uav2/joint_uav_x	j_uav2_x
uav1/joint_uav_y	j_uav1_y
uav2/joint_uav_y	j_uav2_y
uav1/joint_load_x	j_bar1_x
uav2/joint_load_x	j_bar2_x
bar/joint_load_y_1	j_bar1_y
bar/joint_load_y_2	j_bar2_y

Table 5.1: Joint names used in the URDF model and in the launchfiles

Testing the model against disturbances

In order to reproduce these disturbances in Gazebo in a repeatable manner, three separate launchfiles were created: `test_1.launch`, `test_2.launch` and `test_3.launch`, respectively meant to induce $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$.

In each of these tests, the URDF model is spawned in a configuration outside of the equilibrium set. Specifically, the initial pose of the load is the source of the disturbance, while the UAVs are placed in a the neighborhood of their target position. This is done to prevent any disturbance other than the one we are specifically trying to induce.

The exact initial configurations of the model in the three tests are described in the following excerpts of code. Note that the joint names are different from those used in the URDF model: the correspondence is shown in Table 5.1.

The initial conditions described in the launchfiles are represented in Figures 5.10, 5.11 and 5.12 for `test_1.launch`, for `test_2.launch` and `test_3.launch` respectively.

test_1.launch

```
69 <!-- Initial joint values for the robot -->
70 <arg name="j_uav1_x" value="-0.5"/>
71 <arg name="j_uav2_x" value="-0.5"/>
72 <arg name="j_bar1_x" value="0.5"/>
73 <arg name="j_bar2_x" value="0.5"/>
74 <arg name="j_uav1_y" value="0.0"/>
75 <arg name="j_uav2_y" value="0.0"/>
76 <arg name="j_bar1_y" value="0.0"/>
77 <arg name="j_bar2_y" value="0.0"/>
```

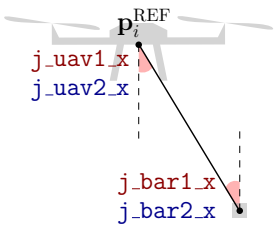


Figure 5.10: Initial conditions in `test_1.launch`, right side view

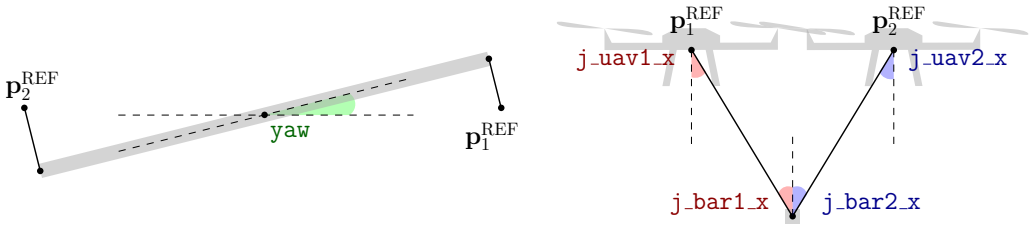
test_2.launch

```
69 <!-- Initial joint values for the robot -->
70 <arg name="j_uav1_x" value="0.0"/>
71 <arg name="j_uav2_x" value="0.0"/>
72 <arg name="j_bar1_x" value="0.0"/>
73 <arg name="j_bar2_x" value="0.0"/>
74 <arg name="j_uav1_y" value="-0.5"/>
75 <arg name="j_uav2_y" value="-0.5"/>
76 <arg name="j_bar1_y" value="0.5"/>
77 <arg name="j_bar2_y" value="0.5"/>
```

Figure 5.11: Initial conditions in `test_2.launch`, front view

test_3.launch

```
67 <arg name="yaw" value="0.36"/>
68
69 <!-- Initial joint values for the robot -->
70 <arg name="j_uav1_x" value="-0.5"/>
71 <arg name="j_uav2_x" value="0.5"/>
72 <arg name="j_bar1_x" value="0.5"/>
73 <arg name="j_bar2_x" value="-0.5"/>
74 <arg name="j_uav1_y" value="0.0"/>
75 <arg name="j_uav2_y" value="0.0"/>
76 <arg name="j_bar1_y" value="0.0"/>
77 <arg name="j_bar2_y" value="0.0"/>
```



(a) Top view (b) Side view

Figure 5.12: Initial conditions in `test_3.launch`

Chapter 6

Control strategy

In this chapter we use insight from the Gazebo simulations to design and test a controller, with an emphasis on counteracting the swinging motion of the load. Two oscillation damping methods are proposed and put to comparison.

6.1 Control based on the position errors of the vehicles

In Chapter 4 we concluded that the system could be stabilized at any point

$$\mathbf{x}^{\mathrm{REF}} = \begin{bmatrix} \mathbf{p}^{\mathrm{REF}} \\ 0 \\ \mathbf{n}_B^{\mathrm{REF}} \\ 0 \\ \mathbf{p}_1^{\mathrm{REF}} \\ 0 \\ \mathbf{p}_2^{\mathrm{REF}} \\ 0 \end{bmatrix} \in \Gamma^{\parallel} \subset \Omega \tag{6.1}$$

by a standard PD control on the position error of each vehicle,

$$\mathbf{e}_{\mathbf{p}_i} = \mathbf{p}_i^{\mathrm{REF}} - \mathbf{p}_i \qquad i = 1, 2 \tag{6.2}$$

$$\dot{\mathbf{e}}_{\mathbf{p}_i} = -\mathbf{v}_i \qquad i = 1, 2\,. \tag{6.3}$$

However, the linearization approach is limited by the fact that the asymptotic stability is only proven for an unknown neighborhood of the equilibrium point. It provides no measurement on the entity of disturbances that the controller can handle, not to mention the fact that it does not guarantee that the vehicles are capable of transporting the load along a trajectory.

Therefore, the next step is to test the performances of the controller in the Gazebo environment, specifically using the previously discussed launchfiles `test_1.launch`, `test_2.launch` and `test_3.launch`.

Note that the controller undergoing the tests is a revised version of (4.15), with the addition of an integral gain in order to remove the steady state error in the altitude:

$$\mathbf{u}_{\mathrm{PID}}(\mathbf{e}_{\mathbf{p}_i}) = \mathrm{K}_P\, \mathbf{e}_{\mathbf{p}_i} + \mathrm{K}_I \int \mathbf{e}_{\mathbf{p}_i} + \mathrm{K}_D\, \dot{\mathbf{e}}_{\mathbf{p}_i} \tag{6.4}$$

with matrix gains

$$\mathrm{K}_P = \begin{bmatrix} k_{Px} & 0 & 0 \\ 0 & k_{Py} & 0 \\ 0 & 0 & k_{Pz} \end{bmatrix} \qquad \mathrm{K}_I = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & k_{Iz} \end{bmatrix} \qquad \mathrm{K}_D = \begin{bmatrix} k_{Dx} & 0 & 0 \\ 0 & k_{Dy} & 0 \\ 0 & 0 & k_{Dz} \end{bmatrix}.$$

In the simulations launched by `test_1.launch`, `test_2.launch` and `test_3.launch`, the system exhibits unstable behavior, with both the load and the vehicles unable to converge to the desired configuration. The error measurements are shown in Figures 6.4, 6.5 and 6.6, at the end of the chapter.

6.2 Oscillations damping methods

The tests show that a controller only based on the position error of the vehicles is insufficient when the load undergoes significant disturbances. It is apparent that a better control strategy requires a method to compensate the oscillations of the bar.

In this thesis, two different solutions are proposed. Albeit distinct, they both stem from the same key insight: an error in the pose of the load (including any undesired swinging motion) can be described as an error in the position and velocity of the anchoring points $\mathbf{p}_{B1}$ and $\mathbf{p}_{B2}$.

Method 1: Compensating the velocities $\mathbf{v}_{Bi}$ of the anchoring points

Let us think of two ends of the bar as two separate swinging objects: while this is not entirely accurate (the dynamics of points $\mathbf{p}_{B1}$ and $\mathbf{p}_{B2}$ in general are not decoupled), it helps in understanding the intuition behind this method.

Indeed, if we consider $\mathbf{p}_{Bi}$ to be a point mass attached to a moving pivot $\mathbf{p}_i$ through a rigid link, then any unwanted motion of the suspended mass can be compensated by moving the pivot accordingly. The concept is the same behind the stabilization of a pendulum with a moving pivot, and is applied every day in construction sites around the world when lifting heavy loads with a crane.

The control law must therefore contain a term that makes each UAV follow the motion of the respective anchoring point, as in Figure 6.1:

$$\mathbf{u}_{\mathrm{OD}}(\mathbf{v}_{Bi}) = \mathrm{K}_V \, \mathbf{v}_{Bi} \tag{6.5}$$

where K_V is a diagonal matrix

$$\mathrm{K}_V = \begin{bmatrix} k_{Vx} & 0 & 0 \\ 0 & k_{Vy} & 0 \\ 0 & 0 & k_{Vz} \end{bmatrix}.$$

Method 2: Correcting the misalignment between $\mathbf{p}_i$ and $\mathbf{p}_{Bi}$

When the load is oscillating, one or both of the anchoring points are not aligned with the respective UAV position, resulting in $\mathbf{n}_{Ci} \neq \mathbf{e}_3$.

Then, a possible control approach consists in making each UAV follow the respective anchoring point, in an attempt to correct the misalignment. Let the error and its time derivative be defined as:

$$\mathbf{e}_{\mathbf{p}_{Bi}} = \mathbf{p}_{Bi} + L_i \mathbf{e}_3 - \mathbf{p}_i \qquad i = 1, 2 \tag{6.6}$$

$$\dot{\mathbf{e}}_{\mathbf{p}_{Bi}} = \mathbf{v}_{Bi} - \mathbf{v}_i \qquad i = 1, 2 \tag{6.7}$$

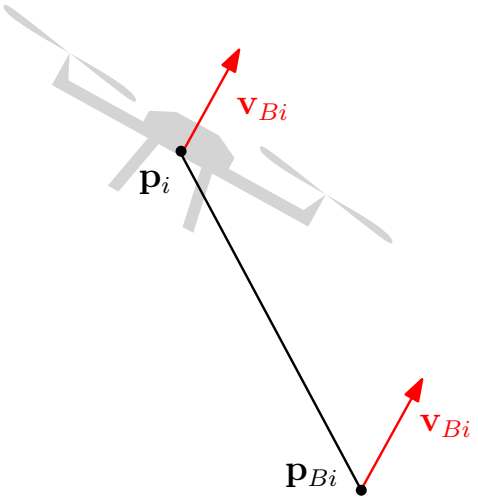


Figure 6.1: Compensating $\mathbf{v}_{Bi}$

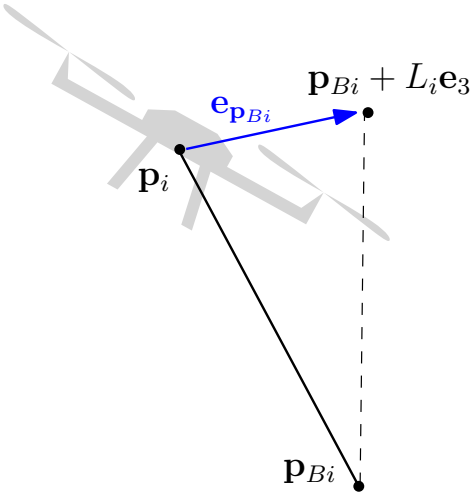


Figure 6.2: Realigning $\mathbf{p}_i$ with $\mathbf{p}_{Bi}$

then the oscillations are damped through a PD controller which forces the UAV to realign with the respective anchoring point, as in Figure 6.2,

$$\mathbf{u}_{\mathrm{OD}}(\mathbf{e}_{\mathbf{p}_{Bi}}) = \mathrm{C}_P\, \mathbf{e}_{\mathbf{p}_{Bi}} + \mathrm{C}_D\, \dot{\mathbf{e}}_{\mathbf{p}_{Bi}} \tag{6.8}$$

where C_P and C_D are diagonal matrices

$$\mathrm{C}_P = \begin{bmatrix} c_{Px} & 0 & 0 \\ 0 & c_{Py} & 0 \\ 0 & 0 & c_{Pz} \end{bmatrix} \qquad \mathrm{C}_I = \begin{bmatrix} c_{Dx} & 0 & 0 \\ 0 & c_{Dy} & 0 \\ 0 & 0 & c_{Dz} \end{bmatrix}.$$

6.3 Implementation of the controller

The controller is implemented with a distributed approach. Each vehicle runs the controller independently from the other, and only needs information related to its own odometry

$$\mathcal{O}_i = \begin{bmatrix} \mathbf{p}_i \\ \mathbf{v}_i \end{bmatrix}$$

and the odometry of the load

$$\mathcal{O} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{n}_B \\ \boldsymbol{\omega}_B \end{bmatrix}.$$

Notice how, with this choice of notation, the state of the system defined in (2.6) can be rewritten as

$$\mathbf{x} = \begin{bmatrix} \mathcal{O} \\ \mathcal{O}_1 \\ \mathcal{O}_2 \end{bmatrix}.$$

The reference trajectory consists only in a series of poses for the load, each described by the pair $\mathbf{p}^{\text{REF}}$ and $\mathbf{n}_B^{\text{REF}}$, rather than the full state

$$\mathbf{x}^{\text{REF}} = \begin{bmatrix} \mathbf{p}^{\text{REF}} \\ 0 \\ \mathbf{n}_B^{\text{REF}} \\ 0 \\ \mathbf{p}_1^{\text{REF}} \\ 0 \\ \mathbf{p}_2^{\text{REF}} \\ 0 \end{bmatrix} \in \Gamma^{\parallel} . \tag{6.1 revisited}$$

As a matter of fact, for any state in $\Gamma \supset \Gamma^{\parallel}$, there is a one to one correspondence between the pose of the load and the vehicle positions: given the reference pose of the bar, then the corresponding vehicle positions are

$$\mathbf{p}_i^{\text{REF}} = \mathbf{p}^{\text{REF}} + d_i \mathbf{n}_B^{\text{REF}} + L_i \mathbf{e}_3 \qquad i = 1, 2 . \tag{6.9}$$

and vice versa, given a pair of UAV positions, the pose of the load is uniquely identified by

$$\begin{cases} \mathbf{n}_B^{\text{REF}} = \dfrac{\mathbf{p}_1^{\text{REF}} - \mathbf{p}_2^{\text{REF}} - (L_1 - L_2)\ \mathbf{e}_3}{d_1 - d_2} & \text{(6.10a)} \\ \mathbf{p}^{\text{REF}} = \mathbf{p}_i^{\text{REF}} - L_i\, \mathbf{e}_3 - d_i\, \mathbf{n}_B^{\text{REF}} \qquad i = 1, 2 . & \text{(6.10b)} \end{cases}$$

Control law

Given the aforementioned odometry measurements and the target position $\mathbf{p}_i^{\text{REF}}$, each UAV is controlled according to

$$\mathbf{u}_{\text{UAV}}(\mathcal{O}, \mathcal{O}_i) = \mathbf{u}_{\mathbf{eq},i} + \mathbf{u}_{\text{PID}}(\mathbf{p}_i^{\text{REF}}, \mathcal{O}_i) + \mathbf{u}_{\text{OD}}(\mathcal{O}, \mathcal{O}_i) . \tag{6.11}$$

The proposed control law is made up of three terms. The equilibrium term $\mathbf{u}_{\mathbf{eq},i}$ is responsible for generating enough thrust to lift the system. It is a constant and is defined according to (4.9) or (4.10) for the first and second UAV respectively.

The position controller $\mathbf{u}_{\text{PID}}$, defined in (6.4), is tasked with generating an input that brings the UAV from its current position $\mathbf{p}_i$ to the target one $\mathbf{p}_i^{\text{REF}}$.

Last, the oscillation damping term $\mathbf{u}_{\text{OD}}$ is a function of the odometry measurements, and can either be (6.5) or (6.8).

Remapping of $\mathbf{n}_B^{\text{REF}}$ and collision avoidance

The Gazebo simulations show that the controller, as it is defined in (6.11), is incapable of properly handling big differences between the current and desired orientation of the load. Specifically, high values of

$$\left| \psi^{\text{REF}} - \psi \right| ,$$

which involve a yaw rotation of the bar, can lead to the UAVs moving towards each other as if trying to switch places, rather then following along a safe circular trajectory.

Therefore, before determining the target positions $\mathbf{p}_1^{\text{REF}}$ and $\mathbf{p}_2^{\text{REF}}$, the difference between current and desired bar orientation is verified, and if necessary $\mathbf{n}_B^{\text{REF}}$ is recomputed

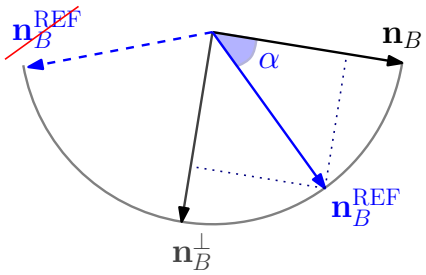


Figure 6.3: Graphical representation of (6.12)

according to

$$\mathbf{n}_B^{\text{REF}} = \begin{cases} \mathbf{n}_B^{\text{REF}} & \text{if } \mathbf{n}_B^{\intercal}\mathbf{n}_B^{\text{REF}} \geq \cos\alpha \\ \mathbf{n}_B & \text{if } \mathbf{n}_B^{\intercal}\mathbf{n}_B^{\text{REF}} \leq -0.99 \\ \cos\alpha\,\mathbf{n}_B + \sin\alpha\,\mathbf{n}_B^{\perp} & \text{else} \end{cases} \tag{6.12}$$

where α is the maximum angle allowed between the current value of $\mathbf{n}_B$ and the desired value $\mathbf{n}_B^{\text{REF}}$.

The algorithm, which is also represented in Figure 6.3, forces the vehicles to follow intermediate trajectories. Indeed, if the angle between $\mathbf{n}_B$ and $\mathbf{n}_B^{\text{REF}}$ is greater than α, then $\mathbf{n}_B^{\text{REF}}$ gets remapped to a unit vector whose angle with $\mathbf{n}_B$ is exactly α; otherwise, if the angle is lesser than α, $\mathbf{n}_B^{\text{REF}}$ remains unchanged.

Note that (6.12) rejects angles close to $\pm\pi$, since the controller in this case cannot decide whether to move clockwise or counterclockwise, leading to an ambiguity.

Comparison of the damping methods

The performances of the control law (6.11) are tested in the Gazebo simulator, for both the oscillation damping methods. The results of `test_1.launch`, `test_2.launch` and `test_3.launch` show that the propose controller is capable of stabilizing the system against disturbances. The error measurements are shown in Figures 6.7, 6.8 and 6.9.

Specifically, the second method appears to be overall faster and more reliable. Further simulations confirm that it also fares better in following trajectories.

6.4 Plots of the test results

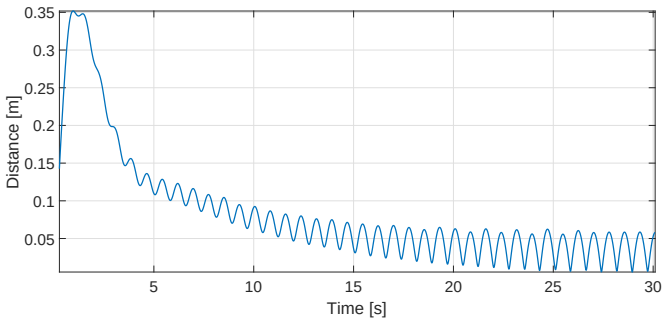
We close this chapter with the plots featuring the error measurements recorded when testing the controllers. For each test, we show four plots: one for the position error of each UAV, and two for the load.

In particular, note that the error on the pose of the load is described in terms of $\mathbf{p}_{B1}$ and $\mathbf{p}_{B2}$, rather than $\mathbf{p}$ and $\mathbf{n}_B$, as the resulting plots are arguably more readable. The goal is to check whether or not the bar converges to the desired pose, and for any trajectory $\mathbf{x}(t) \in \Omega$

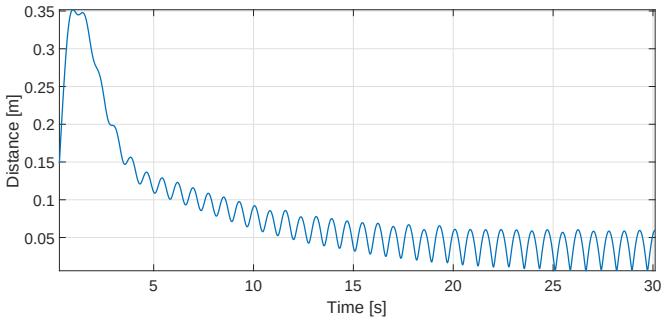
$$\begin{cases} \mathbf{p}(t) \longrightarrow \mathbf{p}^{\text{REF}} \\ \mathbf{n}_B(t) \longrightarrow \mathbf{n}_B^{\text{REF}} \end{cases} \iff \begin{cases} \mathbf{p}_{B1}(t) \longrightarrow \mathbf{p}_{B1}^{\text{REF}} \\ \mathbf{p}_{B2}(t) \longrightarrow \mathbf{p}_{B2}^{\text{REF}} \end{cases}$$

where obviously

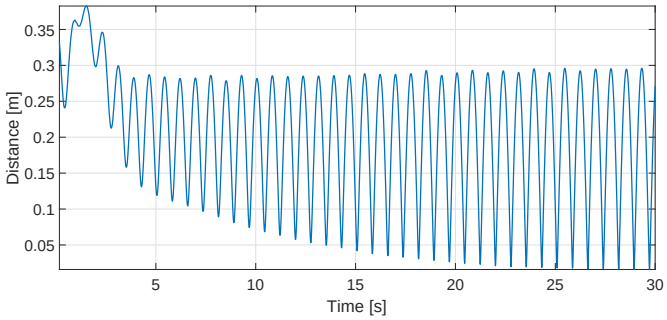
$$\mathbf{p}_{Bi}^{\text{REF}} = \mathbf{p}^{\text{REF}} + d_i\mathbf{n}_B^{\text{REF}} \qquad\qquad i = 1, 2\,.$$



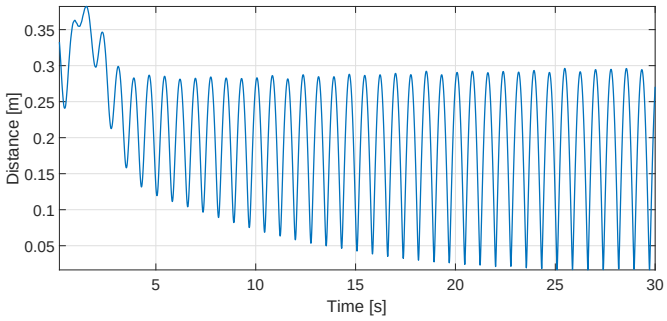
(a) Norm of the position error $\mathbf{e}_{\mathbf{p}_1}$



(b) Norm of the position error $\mathbf{e}_{\mathbf{p}_2}$

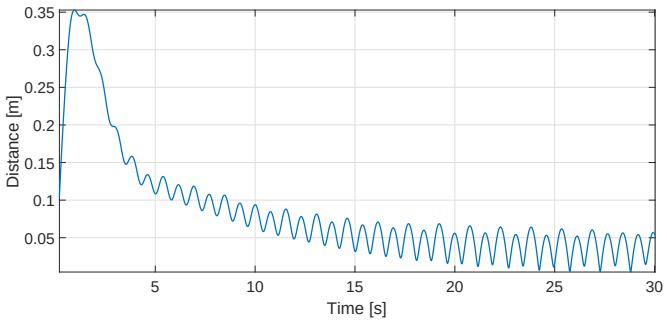


(c) Norm of error $\mathbf{p}_{B1} - \mathbf{p}_{B1}^{\text{REF}}$

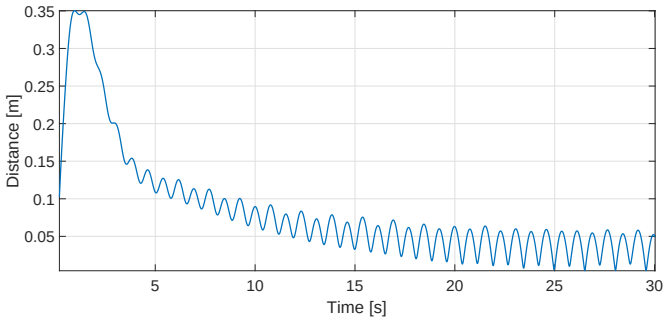


(d) Norm of error $\mathbf{p}_{B2} - \mathbf{p}_{B2}^{\text{REF}}$

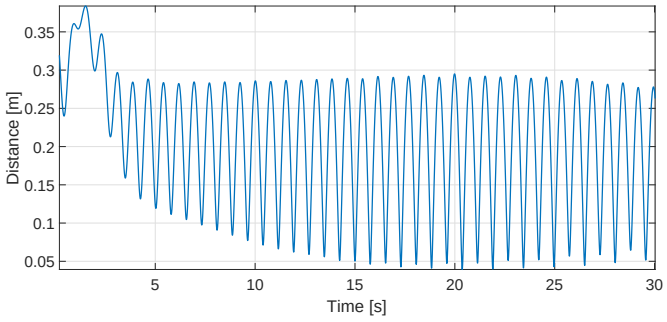
Figure 6.4: Results of `test_1.launch` with PID control on the position error



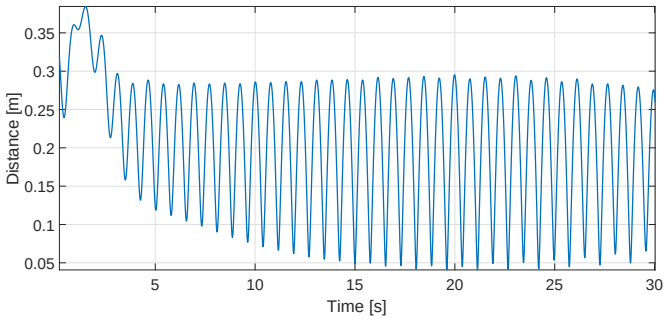
(a) Norm of the position error $\mathbf{e}_{\mathbf{p}_1}$



(b) Norm of the position error $\mathbf{e}_{\mathbf{p}_2}$

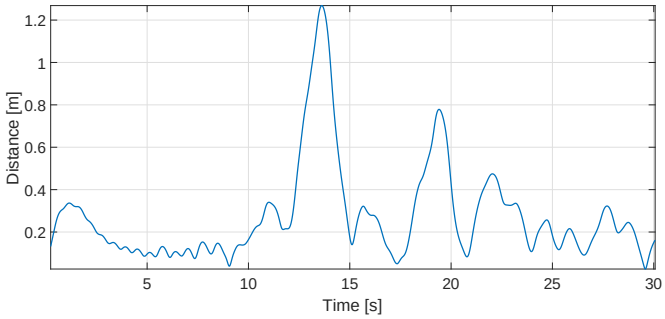


(c) Norm of error $\mathbf{p}_{B1} - \mathbf{p}_{B1}^{\text{REF}}$

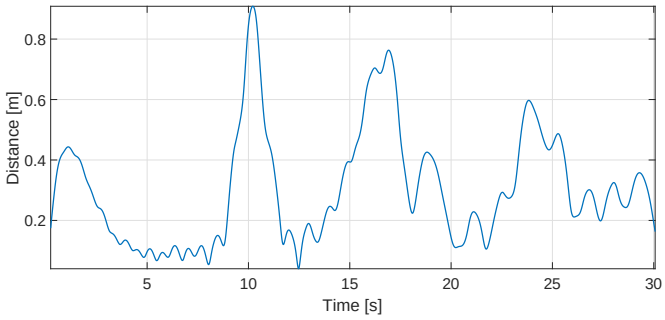


(d) Norm of error $\mathbf{p}_{B2} - \mathbf{p}_{B2}^{\text{REF}}$

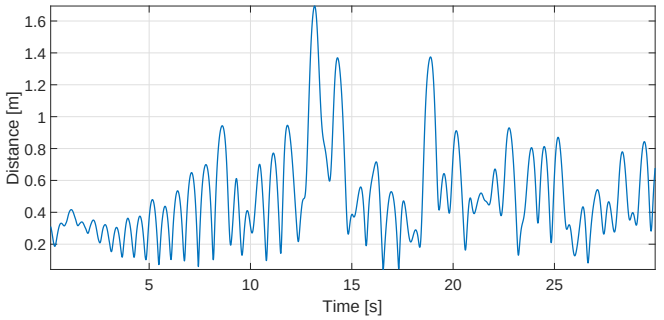
Figure 6.5: Results of `test_2.launch` with PID control on the position error



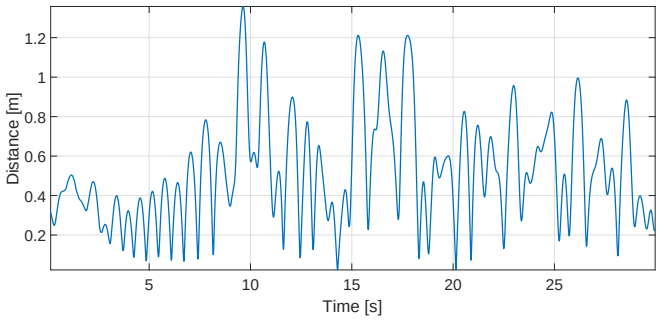
(a) Norm of the position error $\mathbf{e}_{\mathbf{p}_1}$



(b) Norm of the position error $\mathbf{e}_{\mathbf{p}_2}$

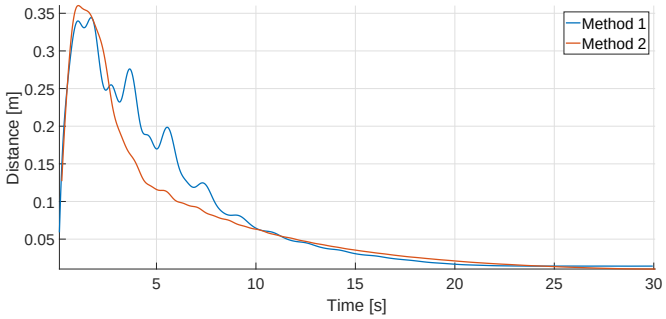


(c) Norm of error $\mathbf{p}_{B1} - \mathbf{p}_{B1}^{\text{REF}}$

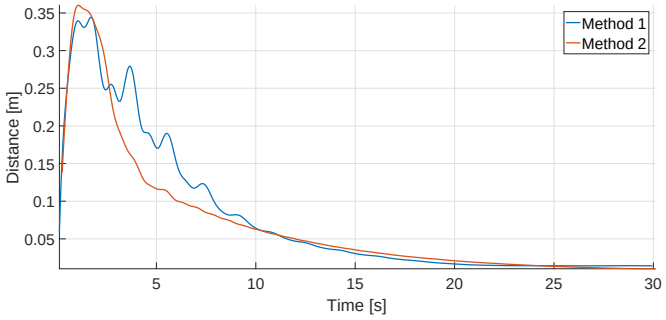


(d) Norm of error $\mathbf{p}_{B2} - \mathbf{p}_{B2}^{\text{REF}}$

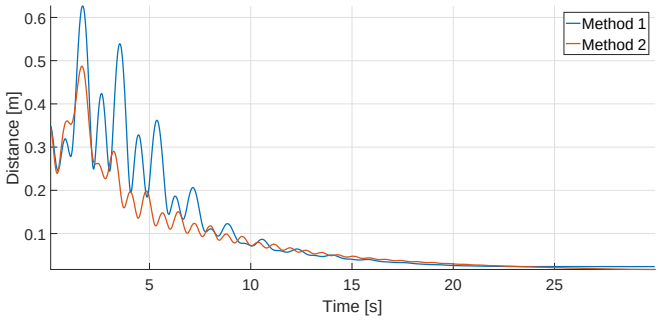
Figure 6.6: Results of `test_3.launch` with PID control on the position error



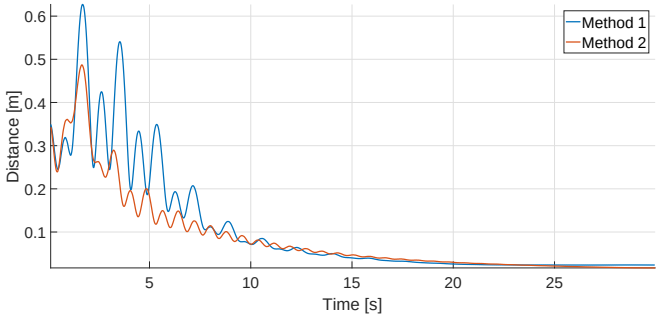
(a) Norm of the position error $\mathbf{e}_{\mathbf{p}_1}$



(b) Norm of the position error $\mathbf{e}_{\mathbf{p}_2}$

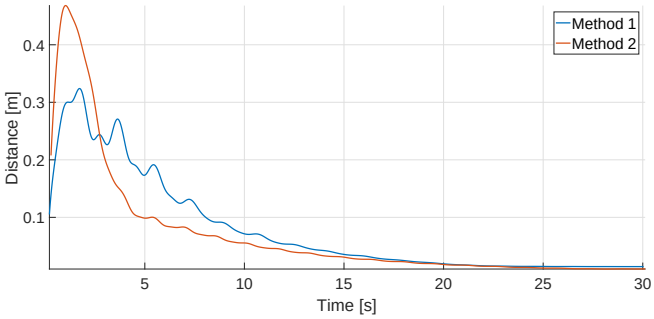


(c) Norm of error $\mathbf{p}_{B1} - \mathbf{p}_{B1}^{\text{REF}}$

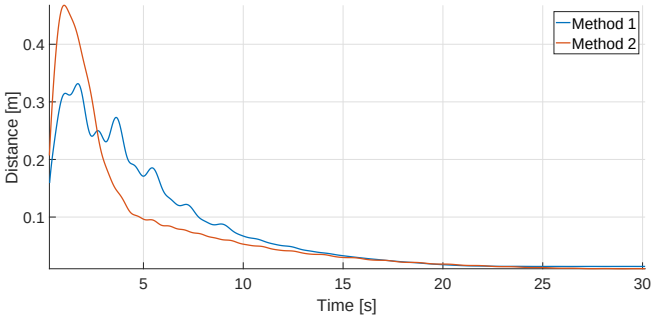


(d) Norm of error $\mathbf{p}_{B2} - \mathbf{p}_{B2}^{\text{REF}}$

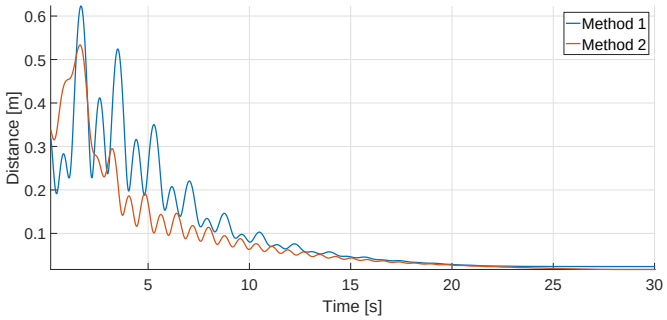
Figure 6.7: `test_1.launch`, comparison between the oscillation damping methods



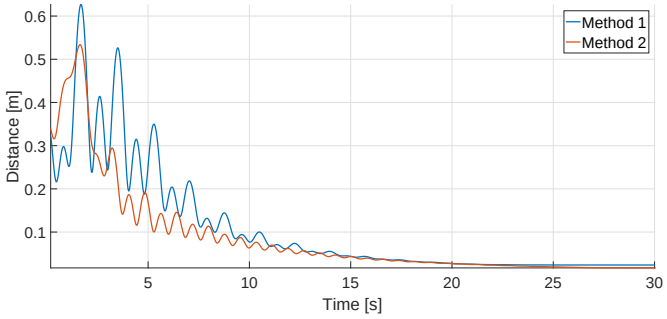
(a) Norm of the position error $\mathbf{e}_{\mathbf{p}_1}$



(b) Norm of the position error $\mathbf{e}_{\mathbf{p}_2}$

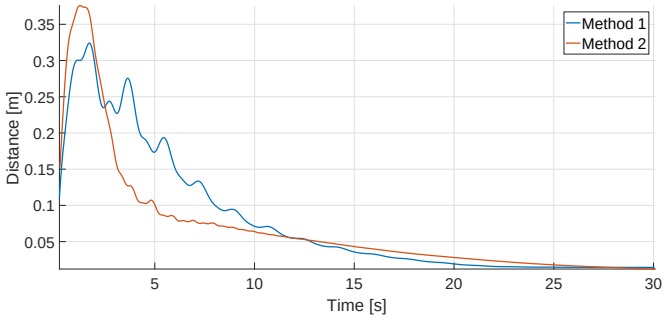


(c) Norm of error $\mathbf{p}_{B1} - \mathbf{p}_{B1}^{\text{REF}}$

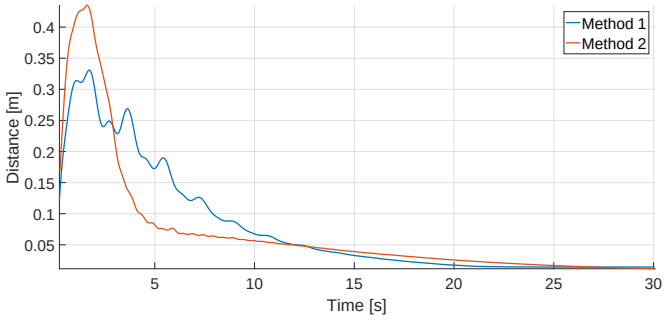


(d) Norm of error $\mathbf{p}_{B2} - \mathbf{p}_{B2}^{\text{REF}}$

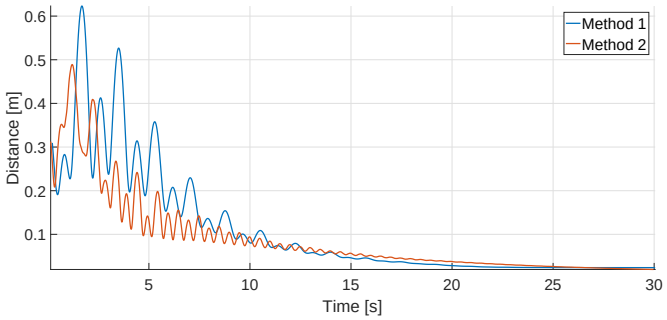
Figure 6.8: `test_2.launch`, comparison between the oscillation damping methods



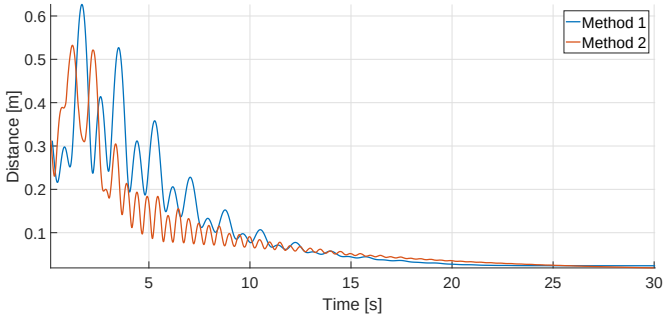
(a) Norm of the position error $\mathbf{e}_{\mathbf{p}_1}$



(b) Norm of the position error $\mathbf{e}_{\mathbf{p}_2}$



(c) Norm of error $\mathbf{p}_{B1} - \mathbf{p}_{B1}^{\text{REF}}$



(d) Norm of error $\mathbf{p}_{B2} - \mathbf{p}_{B2}^{\text{REF}}$

Figure 6.9: `test_3.launch`, comparison between the oscillation damping methods

Chapter 7

Conclusions

In this thesis, we address the problem of collaborative load lifting in two main steps. First, we develop a mathematical model, and use it to understand the system dynamics and isolate a set of safe equilibrium configurations. Then, we bring this model in the Gazebo environment, and use the initial simulation results as a base for developing an improved controller, which is then tested again against disturbances.

We propose two distributed variants of the control law, using different damping methods, which are then compared. The results of the tests on the simulator favor one method over the other, but in general both are shown to be able to achieve convergence to the desired state $\mathbf{x}^{\mathrm{REF}}$, provided that it belongs to a specific subset of equilibrium points,

$$\Gamma^{\parallel} = \{\mathbf{x} \in \Gamma \,:\, \mathbf{n}_B^{\mathsf{T}} \mathbf{e}_3 = 0\} \,. \qquad \text{(4.13 revisited)}$$

Future developments

The natural follow-up of this endeavor involves testing the proposed controller with real life drones. In particular, see how well the oscillation damping capabilities translate into reality.

There are some evident challenges in this new goal. First of all, in the simulator our control law uses exact odometry measurements, which are not always easily achievable in the real world. Even with a motion capture system in place, there is still a degree of uncertainty in the measurements; therefore a first development would involve testing the robustness of the controller, in order to measure how truly reliant it is on the precision of the odometry detection.

Recall that the system is modeled as a bar tethered to two UAVs. However, a difficulty arises when using cable or rope, since the Gazebo model treats the connection between UAV and load as a rigid link. Thus, tests should focus on evaluating if the oscillation damping methods still work when using actual rope. A possible solution may involve combining the results of [5, 6] with our control strategy.

Lastly, yet another direction may involve exploring the situation where the equations of the linear system cannot be decoupled, and see whether our control approach can be generalized.

Appendix

Rotation matrices

Consider a rigid body, and its local reference frame $x'y'z'$. By definition the orientation of $x'y'z'$ (and therefore of the body) is characterized by the coordinates of its standard unit vectors $\mathbf{e}_{x'}$, $\mathbf{e}_{y'}$ and $\mathbf{e}_{z'}$ expressed with respect to the global reference frame xyz:

$$\mathbf{R} = \begin{bmatrix} \mathbf{e}_{x'} & \mathbf{e}_{y'} & \mathbf{e}_{z'} \end{bmatrix} \tag{1}$$

which is called a rotation matrix. Note that, being made up of standard unit vectors, each rotation matrix must satisfy the following constraints:

$$\begin{aligned} &\mathbf{e}_{x'}^{\intercal}\mathbf{e}_{x'} = 1 && \mathbf{e}_{x'}^{\intercal}\mathbf{e}_{y'} = 0 && \mathbf{e}_{x'}^{\intercal}\mathbf{e}_{z'} = 0 \\ &&& \mathbf{e}_{y'}^{\intercal}\mathbf{e}_{y'} = 1 && \mathbf{e}_{y'}^{\intercal}\mathbf{e}_{z'} = 0 \\ &&&&& \mathbf{e}_{z'}^{\intercal}\mathbf{e}_{z'} = 1 \end{aligned}$$

that make the matrix orthogonal. In particular, we only consider rotation matrices that belong to the special orthonormal group:

$$\mathbf{R}^{\intercal} = \mathbf{R}^{-1} \qquad\qquad \det(\mathbf{R}) = 1\,,$$

where the positivity of the determinant is linked to the frame being right handed.

Elementary rotations

An elementary rotation occurs when $x'y'z'$ only rotates about one of the axes of the world reference frame xyz:

$$\mathbf{R}_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \tag{2}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3}$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Three angle representation

A rotation matrix gives a redundant description of orientation. It has nine elements and needs to satisfy six constraints, leaving three free parameters.

A representation of orientation in terms of three independent parameters constitutes a minimal representation. It can be obtained using a set of three Euler angles (and therefore three elementary rotations), according to the following theorem.

Euler's Theorem. *Any two independent orthonormal reference frames xyz and $x'y'z'$ can be related by a sequence of no more than three elementary rotations, as long as no two successive rotations are about the same axis.*

Roll-Pitch-Yaw convention

In this thesis, we use the following sequence of rotations as a convention:

1. a rotation of and angle φ around the x axis of the world reference frame
2. a rotation of and angle θ around the y axis of the world reference frame
3. a rotation of and angle ψ around the z axis of the world reference frame.

The resulting rotation is obtained through pre-multiplication

$$\mathbf{R} = \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\varphi) \tag{5}$$

since the rotations are with respect to a fixed reference frame.

Bibliography

[1] T. Lee, K. Sreenath, and V. Kumar, *Geometric Control of Cooperating Multiple Quadrotor UAVs with a Suspended Payload*, in Proceedings of the IEEE Conference on Decision and Control, vol. 5510-5515, Florence, Italy, Dec. 2013.

[2] T. Lee, *Geometric Control of Multiple Quadrotor UAVs Transporting a Cable-Suspended Rigid Body*, Proceedings of the IEEE Conference on Decision and Control, pp. 6155-6160, Dec. 2014.

[3] N. Michael, J. Fink, and V. Kumar, *Cooperative Manipulation and Transportation with Aerial Robots*, in Proc. Robotic Science and Systems, 2009.

[4] K. Sreenath, N. Michael, and V. Kumar *Trajectory Generation and Control of a Quadrotor with a Cable-Suspended Load - A Differentially-Flat Hybrid System*, in IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, pp. 4873-4880, May 2013.

[5] P. Pereira, M. Herzog, and D. Dimarogonas, *Slung Load Transportation with a Single Aerial Vehicle and Disturbance Removal*, in Mediterranean Conference on Control and Automation, pp. 671-676, 2016.

[6] P. Pereira, and D. Dimarogonas, *Control Framework for Slung Load Transportation with Two Aerial Vehicles*, manuscript, 2017.

[7] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*, Springer International Publishing, 2016.

[8] `http://wiki.ros.org/urdf/XML/link`

[9] `http://wiki.ros.org/urdf/XML/joint`