UNIVERSITY of PADOVA
Dipartimento di Ingegneria dell'Informazione

Laurea Magistrale in Control Systems Engineering

Robotics and Control 2
Lecture notes

A. Cenedese, the SPARCS Group
and the class of 2016-17

Version: December 11, 2023

Academic year 2023-2024

Contents

Note to the reader: *in these lecture notes some concepts and notions (highlighted in light blue color) may appear as ancillary to the whole discussion, but they are important for completeness of thought and for the full understanding of the subject. Nonetheless, the student can harmlessly skip these parts during the first round of reading and leave them for a second moment of further (deeper) study.*

Chapter 1

Mobile agent systems

1.1 Algebraic groups of interest

Definition 1 (Algebraic group) *An algebraic group is an algebraic structure equipped with a binary operation ($\circ$) and characterized by the following four properties:*

1. ***Closure*** *Group $\mathcal{G}$ is close:*
$$\forall\, g_1, g_2 \in \mathcal{G}: \quad g_1 \circ g_2 \in \mathcal{G} \tag{1.1}$$
2. ***Associativity*** *Group $\mathcal{G}$ is associative:*
$$\forall\, g_1, g_2, g_3 \in \mathcal{G}: \quad g_1 \circ (g_2 \circ g_3) = (g_1 \circ g_2) \circ g_3 \tag{1.2}$$
3. ***Identity*** *There exists the neutral element*
$$\exists 1 \in \mathcal{G} \quad \textit{such that} \quad g \circ 1 = 1 \circ g = g \tag{1.3}$$
4. ***Invertibility*** *There exists the inverse element*
$$\exists\, g^{-1} \in \mathcal{G} \quad \textit{such that} \quad g \circ g^{-1} = g^{-1} \circ g = 1 \tag{1.4}$$

Now it is presented a list of different algebraic groups of interest, with some notes regarding their meaning.

Definition 2 (General linear group $\mathbb{GL}(n)$) *(of dimension n)*
$\mathbb{GL}(n)$ is the set of all the linear transformations/matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$ that are non-singular and equipped with the standard matricial product.

$$\begin{aligned} \mathbf{A} : \mathbb{R}^n \quad &\rightarrow \ \mathbb{R}^n \\ \mathbf{x} \quad &\rightarrow \ \mathbf{A}\mathbf{x} \end{aligned} \tag{1.5}$$

Definition 3 (Affine group $\mathbb{A}(n)$) *(of dimension n)*
$\mathbb{A}(n)$ is defined by matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and vector $\mathbf{b} \in \mathbb{R}^n$ with the map:

$$\begin{aligned} \mathbf{A} : \mathbb{R}^n \quad &\rightarrow \mathbb{R}^n \\ (\mathbf{A}, \mathbf{b}) : \mathbf{x} \quad &\rightarrow \mathbf{A}\mathbf{x} + \mathbf{b} \end{aligned} \tag{1.6}$$

Remark 1 *We can do an embedding in $\mathbb{R}^{n+1}$ in order to obtain an homogeneous representation of $\mathbf{x}$:*

$$\underbrace{\mathbf{x}}_{\mathbb{R}^n} \rightarrow \underbrace{\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}}_{\mathbb{R}^{n+1}} \tag{1.7}$$

$$\mathbf{x} \rightarrow \mathbf{A}\mathbf{x} + \mathbf{b} \quad \Rightarrow \quad \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathbf{x} + \mathbf{b} \\ 1 \end{bmatrix} \tag{1.8}$$

So, the application of an affine transformation to vector $\mathbf{x}$ becomes a matrix product in homogeneous coordinates, where $\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 1 \end{bmatrix} \in \mathbb{GL}(n+1)$.
Embedding in an higher dimensional space is useful to obtain a linear transformation as in the following:

$$\mathbf{x} \overset{(\mathbf{A}_1,\mathbf{b}_1)}{\longrightarrow} \mathbf{A}_1\mathbf{x}+\mathbf{b}_1 \overset{(\mathbf{A}_2,\mathbf{b}_2)}{\longrightarrow} \mathbf{A}_2(\mathbf{A}_1\mathbf{x}+\mathbf{b}_1)+\mathbf{b}_2 = \mathbf{A}_2\mathbf{A}_1\mathbf{x}+\mathbf{A}_2\mathbf{b}_1+\mathbf{b}_2 \tag{1.9}$$

$$\begin{bmatrix}\mathbf{x}\\1\end{bmatrix} \to \begin{bmatrix}\mathbf{A}_1 & \mathbf{b}_1\\0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix} \to \begin{bmatrix}\mathbf{A}_2 & \mathbf{b}_2\\0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{A}_1 & \mathbf{b}_1\\0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix} = \begin{bmatrix}\mathbf{A}_2\mathbf{A}_1 & \mathbf{A}_2\mathbf{A}_1+\mathbf{b}_2\\0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix}. \tag{1.10}$$

Definition 4 (Orthogonal group $\mathbb{O}(n)$) *(of dimension n)*
$\mathbb{O}(n)$ is defined by matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ ($\mathbf{A} \in \mathbb{GL}(n)$) invertible such that the inner product between vectors in $\mathbb{R}^n$ is preserved:

$$\forall \mathbf{x},\mathbf{y} \in \mathbb{R}^n : \quad \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{y}\rangle = \langle \mathbf{x},\mathbf{y}\rangle . \tag{1.11}$$

It is the group of orthogonal matrices:

$$\langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{y}\rangle = \mathbf{x}^\top\mathbf{A}^\top\mathbf{A}\mathbf{y} = \mathbf{x}^\top\mathbf{y} = \langle \mathbf{x},\mathbf{y}\rangle \quad \Rightarrow \quad \mathbf{A}^\top\mathbf{A} = \mathbf{I}. \tag{1.12}$$

Remark 2 *We have distance preserving, angle preserving and area preserving transformations, in other words they are a subset of isometries (e.g. rotations, reflections, symmetries).*

Remark 3 *We are interested in a subgroup of $\mathbb{O}(n)$, which is the **special orthogonal group** $\mathbb{SO}(n)$:*

$$\mathbb{SO}(n) = \{\mathbf{A} \in \mathbb{O}(n) \quad \textit{s.t.} \quad \det(\mathbf{A}) = 1\}. \tag{1.13}$$

Note that $\mathbb{SO}(n)$ is a group of rotations (for example $\mathbb{SO}(2)$ describes planar rotations).

Definition 5 (Euclidean group $\mathbb{E}(n)$) *(of dimension n)*
$\mathbb{E}(n)$ is defined by an affine transformation, where $\mathbf{A} \in \mathbb{O}(n), \mathbf{b} \in \mathbb{R}^n$:

$$\begin{aligned} \mathbf{A} : \mathbb{R}^n &\to \mathbb{R}^n \\ \mathbf{x} &\to \mathbf{A}\mathbf{x} + \mathbf{b} \end{aligned} \tag{1.14}$$

Remark 4 *We are interested in a particular subgroup of $\mathbb{E}(n)$, where $\mathbf{R} \in \mathbb{SO}(n)$, called **special euclidean group** $\mathbb{SE}(n)$. Note that $\mathbb{SE}(n)$ is the group of rototranslations, which can be applied through matrix multiplications in (n+1) as in (1.10).*

The orthogonal group is a subgroup of the general linear group, while the euclidean group is a subgroup of the affine group. Summarizing, we have the following relations:

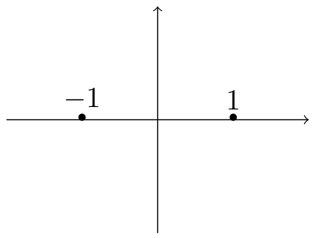
$$\overbrace{\mathbb{SO}(n)}^{\mathbf{R}_+} \subseteq \overbrace{\mathbb{O}(n)}^{\mathbf{R}} \subseteq \overbrace{\mathbb{GL}(n)}^{\mathbf{A}} \tag{1.15}$$

$$\underbrace{\mathbb{SE}(n)}_{(\mathbf{R}_+,\mathbf{T})} \subseteq \underbrace{\mathbb{E}(n)}_{(\mathbf{R},\mathbf{T})} \subseteq \underbrace{\mathbb{A}(n)}_{(\mathbf{A},\mathbf{b})} \subseteq \underbrace{\mathbb{GL}(n+1)}_{\tilde{\mathbf{A}}=\begin{bmatrix}\mathbf{A} & \mathbf{b}\\0 & 1\end{bmatrix}} \tag{1.16}$$

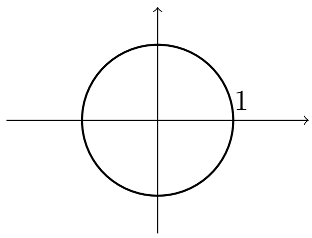
Given these premises, we observe the following relations between algebraic groups and unitary spheres $\mathbb{S}^n$.

$$\mathbb{S}^n = \left\{ x \in \mathbb{R}^{n+1} : \|x\| = 1 \right\} \tag{1.17}$$

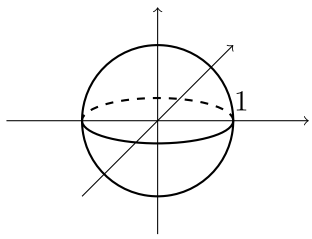
0-sphere : $\mathbb{S}^0 \sim \mathbb{O}(1)$



1-sphere : $\mathbb{S}^1 \sim \mathbb{SO}(2) \sim \mathbb{SO}(2)/\mathbb{SO}(1)$



2-sphere : $\mathbb{S}^2 \sim \mathbb{SO}(3)/\mathbb{SO}(2)$



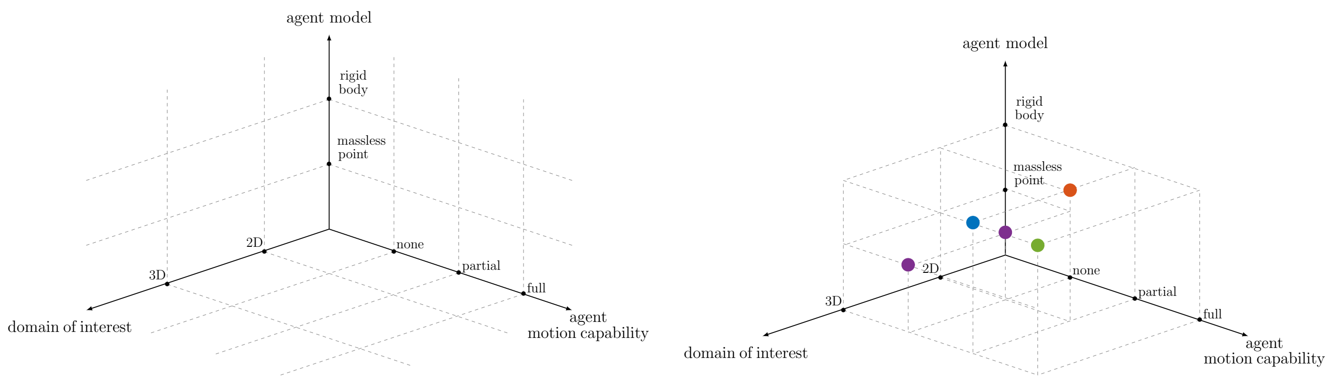
1.2 Pose representation for multi-agent systems

The characteristics of multi-agent systems can be grossly summarized into three different classes (three Cartesian axes in Fig. 1.1(a)):

- one axis takes cares of the agent model distinguishing between massless point and rigid body,
- a second axis accounts for the dimension of the domain of interest that can be a two-dimensional or a three-dimensional space,
- a third axis considers the agent motion capability.

Note that the chosen agent model determines the number of parameters necessary to describe the device location. In particular, with reference to Fig. 1.1(b),

- violet markers: typical model for sensor networks, characterized by a massless entities in the 2D or 3D space with no actuation/motion capability;
- orange and blue markers: camera networks modeled as a set of rigid bodies having partial motion capability and acting in the 2D or 3D space, respectively;
- green marker: aerial vehicles, modeled as a rigid body having full motion capability and acting in the 3D space.



(a) Possible general feature characterization

(b) Actual systems of interest

Figure 1.1: Multi-agent system classification.

We have also to distinguish between the domain of the *state* of the agents and that of the *measurement* of the agents, as in the following Tab. 1.2.

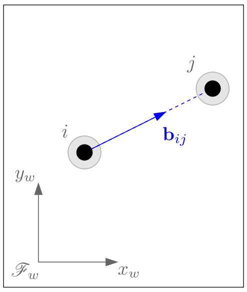
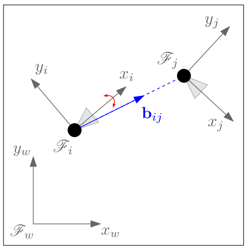
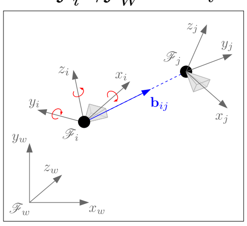
System example	Agent domain	Measurements
massless point system 	$\mathbb{R}^n$ $\mathbf{p}_j$	$\begin{cases} \text{distance} = \mathbb{R} \\ \text{bearing} = \mathbb{S}^{n-1} \end{cases}$ $\quad d_{ij} = \lVert \mathbf{p}_i - \mathbf{p}_j \rVert$, $\mathbf{b}_{ij} = \dfrac{\mathbf{p}_j - \mathbf{p}_i}{\lVert \mathbf{p}_i - \mathbf{p}_j \rVert}$ DOFs = n
planar system $\mathbf{R}_{\mathcal{F}_i \to \mathcal{F}_W} = \mathbf{R}_i$ 	$\mathbb{SE}(2) = \mathbb{R}^2 \times \mathbb{SO}(2)$ $(\mathbf{p}_j, \alpha_j)$	$\begin{cases} \text{distance} = \mathbb{R} \\ \text{bearing} = \mathbb{S}^{1} \end{cases}$ $\quad d_{ij} = \lVert \mathbf{p}_i - \mathbf{p}_j \rVert$, $\mathbf{b}_{ij} = \mathbf{R}_i^\top \dfrac{\mathbf{p}_j - \mathbf{p}_i}{\lVert \mathbf{p}_i - \mathbf{p}_j \rVert}$ DOFs = 2+1 = 3
full 3D system $\mathbf{R}_{\mathcal{F}_i \to \mathcal{F}_W} = \mathbf{R}_i$ 	$\mathbb{SE}(3) = \mathbb{R}^3 \times \mathbb{SO}(3)$ $(\mathbf{p}_j, \mathbf{R}_j)$	$\begin{cases} \text{distance} = \mathbb{R} \\ \text{bearing} = \mathbb{S}^{2} \end{cases}$ $\quad d_{ij} = \lVert \mathbf{p}_i - \mathbf{p}_j \rVert$, $\mathbf{b}_{ij} = \mathbf{R}_i^\top \dfrac{\mathbf{p}_j - \mathbf{p}_i}{\lVert \mathbf{p}_i - \mathbf{p}_j \rVert}$ DOFs = 3+3 = 6

Table 1.1: Pose (attitude and position) and measurement domains for some systems of interest.

1.2.1 Attitude representation: rotation matrices and Euler angles

A common way to represent the attitude of an object or better the relative orientation between a frame $\mathcal{F}_W$ (usually, the fixed inertial world frame) and the frame $\mathcal{F}_B$ centered on the center of mass of the object of interest (body frame) is the rotation matrix, namely a matrix $\mathbf{R} \in \mathbb{SO}(n)$, where $n = 2$ if the rotation is planar and $n = 3$ if it is a full 3D rotation. See Fig. 1.2.

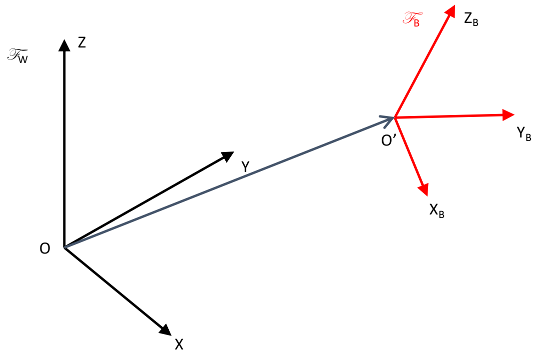


Figure 1.2: Right-handed 3D world and body reference frames, $\mathcal{F}_W$ and $\mathcal{F}_B$.

The translation of $\mathcal{F}_B$ with respect to $\mathcal{F}_W$ is described by the following relation

$$O' = O'_x\mathbf{x} + O'_y\mathbf{y} + O'_z\mathbf{z} = \begin{bmatrix} O'_x \\ O'_y \\ O'_z \end{bmatrix} \tag{1.18}$$

while for the rotation it holds

$$\begin{cases} \mathbf{x}_B = x'_x\mathbf{x} + x'_y\mathbf{y} + x'_z\mathbf{z} \\ \mathbf{y}_B = y'_x\mathbf{x} + y'_y\mathbf{y} + y'_z\mathbf{z} \\ \mathbf{z}_B = z'_x\mathbf{x} + z'_y\mathbf{y} + z'_z\mathbf{z} \end{cases} \tag{1.19}$$

which allow to define the rotation matrix Body2World (B2W) $\mathbf{R}_{B2W} = \mathbf{R}_{WB}$ as the rotation of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$:

$$\mathbf{R}_{WB} = \begin{bmatrix} \mathbf{x}_B & \mathbf{y}_B & \mathbf{z}_B \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} <\mathbf{x}_B, \mathbf{x}> & <\mathbf{y}_B, \mathbf{x}> & <\mathbf{z}_B, \mathbf{x}> \\ <\mathbf{x}_B, \mathbf{y}> & <\mathbf{y}_B, \mathbf{y}> & <\mathbf{z}_B, \mathbf{y}> \\ <\mathbf{x}_B, \mathbf{z}> & <\mathbf{y}_B, \mathbf{z}> & <\mathbf{z}_B, \mathbf{z}> \end{bmatrix} \tag{1.20}$$

We have that a vector $\mathbf{v}_B$ in $\mathcal{F}_B$ rotated through $\mathbf{R}_{B2W}$ w.r.t. $\mathcal{F}_W$ is related to the same vector $\mathbf{v}_W$ in $\mathcal{F}_W$ as

$$\mathbf{v}_W = \mathbf{R}_{B2W}\mathbf{v}_B \tag{1.21}$$

By recalling the $\mathbb{SO}(3)$ nature of rotation matrix, we also have that:

$$\mathbf{v}_B = \mathbf{R}_{B2W}^{-1}\mathbf{v}_W = \mathbf{R}_{B2W}^{\top}\mathbf{v}_W = \mathbf{R}_{W2B}\mathbf{v}_W. \tag{1.22}$$

The rotation matrix description is a redundant description since it has 9 entries when the d.o.f.s of a 3D rotation are just 3.
In the same spirit, it is possible at this stage to write the elementary rotation matrices, considering the rotation around only one axis of the fixed reference frame at a time; the rotations of an angle around a generic axis can be represented by composition of rotation matrices around a chosen frame axes (see Fig. 1.3).

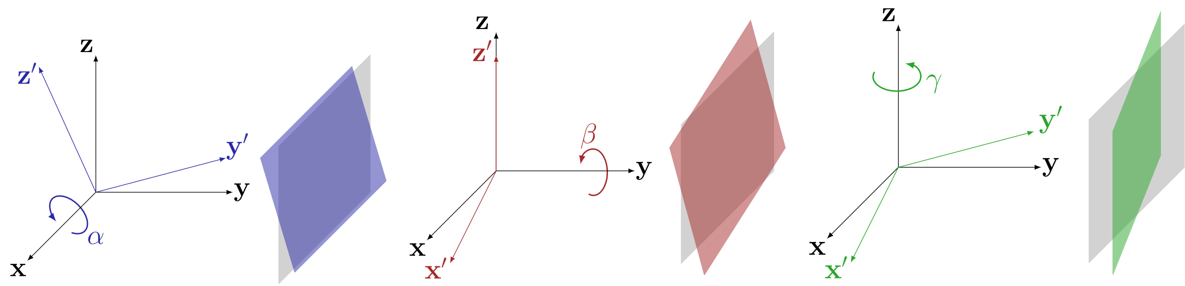


Figure 1.3: Elementary rotations of the 3D reference frame.

If the body reference frame $\mathcal{F}_B$ is rotated with respect to the world frame $\mathcal{F}_W$ of angles (α, β, γ), we define the elementary rotation matrices as

$$\mathbf{R}_x(\alpha) = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{array}\right] \tag{1.23}$$

$$\mathbf{R}_y(\beta) = \left[\begin{array}{c|c|c} \cos\beta & 0 & \sin\beta \\ \hline 0 & 1 & 0 \\ \hline -\sin\beta & 0 & \cos\beta \end{array}\right] \tag{1.24}$$

$$\mathbf{R}_z(\gamma) = \left[\begin{array}{cc|c} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ \hline 0 & 0 & 1 \end{array}\right] \tag{1.25}$$

At the same time, we can choose specific rotations around a coordinate body frame and for a mobile robot we introduce the rotation around z-axis (*yaw* angle ψ), the rotation around y-axis (*pitch* angle θ), and the rotation around x-axis (*roll* angle ϕ).
Similarly, for a camera system we define *pan* as the rotation around x-axis (corresponding to *yaw*) and *tilt* as the rotation around y-axis (corresponding to *pitch*) while the rotation around z-axis is (usually) not considered.
See Fig. 1.4 for a schematic example.

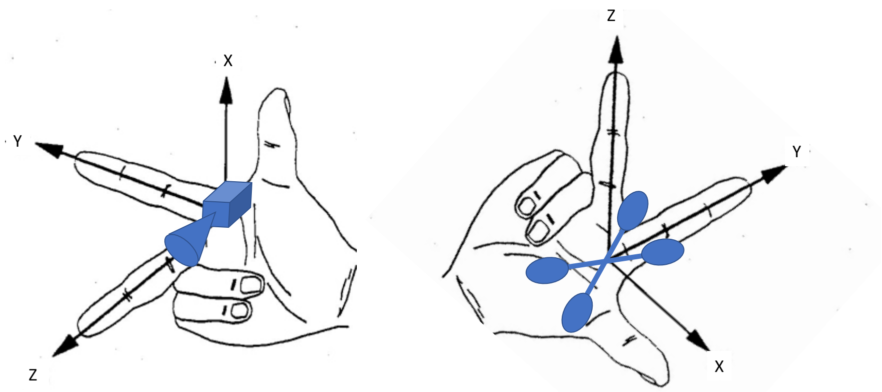


Figure 1.4: Right-handed 3D reference frames w.r.t. a camera (sx) or a UAV (dx).

In particular, we use a specific rotation matrix to describe the attitude of an object in the 3D space (e.g. a UAV) by choosing the order of rotations around the three axes of $\mathcal{F}_{\mathcal{B}}$, namely we rotate around the yaw angle ψ, then the pitch θ, and the roll angle ϕ:

$$\mathbf{R}_{B2W} = \mathbf{R}(\phi, \theta, \psi) = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi) \tag{1.26}$$

where $\mathbf{R}_z(\psi)$, $\mathbf{R}_y(\theta)$ and $\mathbf{R}_x(\phi)$ express the *yaw*, *pitch* and *roll* angles around the z, y and x axes (RPY - XYZ Euler angles).
Finally, by explicitly considering (1.23)-(1.24)-(1.25), formula (1.26) can be expressed by

$$\mathbf{R}_{B2W} = \mathbf{R}(\phi, \theta, \psi) = \left[\begin{array}{c|c|c} \mathrm{c}_\psi \mathrm{c}_\theta & -\mathrm{s}_\psi \mathrm{c}_\phi + \mathrm{c}_\psi \mathrm{s}_\theta \mathrm{s}_\phi & \mathrm{s}_\psi \mathrm{s}_\phi + \mathrm{c}_\psi \mathrm{s}_\theta \mathrm{c}_\phi \\ \mathrm{s}_\psi \mathrm{c}_\theta & \mathrm{c}_\psi \mathrm{c}_\phi + \mathrm{s}_\psi \mathrm{s}_\theta \mathrm{s}_\phi & -\mathrm{c}_\psi \mathrm{s}_\phi + \mathrm{s}_\psi \mathrm{s}_\theta \mathrm{c}_\phi \\ -\mathrm{s}_\theta & \mathrm{c}_\theta \mathrm{s}_\phi & \mathrm{c}_\theta \mathrm{c}_\phi \end{array}\right] \tag{1.27}$$

where c and s is a shorter notation to denote $\cos$ and $\sin$ respectively. This is the *direct transformation* from angles to rotation matrix.

Conversely, when we need to compute how to go from $\mathbf{R}$ to ψ, θ, ϕ we can use the following formulas for retrieve the angles we are looking for. Firstly, define

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{1.28}$$

Secondly, analyze the following three cases.
1) If $\theta \neq \pm\dfrac{\pi}{2}$ then $c_\theta \neq 0$; so, the angle values can be computed as

$$\theta = \arcsin(-r_{31}) \tag{1.29}$$
$$\phi = \operatorname{arctan2}(r_{21}, r_{11}) \tag{1.30}$$
$$\psi = \operatorname{arctan2}(r_{32}, r_{33}) \tag{1.31}$$

where $\operatorname{arctan2}(b, a)$ is the same function used for obtaining the argument of a generic complex number $a + jb$.
2) If $\theta = \dfrac{\pi}{2}$ then $c_\theta = 0$ and $s_\theta = 1$; so, ψ and ϕ values cannot be computed explicitly because the rotation matrix $\mathbf{R}$ does not provide valid arguments for the $\operatorname{arctan2}$ function. Only the information about their difference can be inferred:

$$\mathbf{R} = \begin{bmatrix} \boxed{\begin{matrix} 0 \\ 0 \end{matrix}} & \begin{matrix} -s_\psi c_\phi + c_\psi s_\phi \\ c_\psi c_\phi + s_\psi s_\phi \end{matrix} & \begin{matrix} s_\psi s_\phi + c_\psi c_\phi \\ -c_\psi s_\phi + s_\psi c_\phi \end{matrix} \\ -1 & \multicolumn{2}{c}{\boxed{0 \qquad\qquad 0}} \end{bmatrix} = \left[\begin{array}{c|cc} 0 & s_{\phi-\psi} & c_{\phi-\psi} \\ 0 & c_{\phi-\psi} & -s_{\phi-\psi} \\ \hline -1 & 0 & 0 \end{array}\right]$$

Therefore, we can just compute

$$\phi - \psi = \operatorname{arctan2}(r_{12}, r_{22}) \tag{1.32}$$

3) If $\theta = -\dfrac{\pi}{2}$ then $c_\theta = 0$ and $s_\theta = -1$; so, ψ and ϕ values cannot be computed explicitly because the rotation matrix $\mathbf{R}$ does not provide valid arguments for the $\operatorname{arctan2}$ function. Only the information about their difference can be inferred:

$$\mathbf{R} = \begin{bmatrix} \boxed{\begin{matrix} 0 \\ 0 \end{matrix}} & \begin{matrix} -s_\psi c_\phi - c_\psi s_\phi \\ c_\psi c_\phi - s_\psi s_\phi \end{matrix} & \begin{matrix} s_\psi s_\phi - c_\psi c_\phi \\ -c_\psi s_\phi - s_\psi c_\phi \end{matrix} \\ 1 & \multicolumn{2}{c}{\boxed{0 \qquad\qquad 0}} \end{bmatrix} = \left[\begin{array}{c|cc} 0 & -s_{\phi+\psi} & -c_{\phi+\psi} \\ 0 & c_{\phi+\psi} & -s_{\phi+\psi} \\ \hline 1 & 0 & 0 \end{array}\right]$$

Therefore, we can just compute

$$\phi + \psi = \operatorname{arctan2}(r_{23}, r_{13}) \tag{1.33}$$

Observation 1 *We have seen that for $\theta = \pm\dfrac{\pi}{2}$ we obtain information on $\psi \mp \phi$ only. These are called **gimbal lock** configurations.*

Absolute and relative position

Consider two systems $\mathcal{F}_B^i$, $\mathcal{F}_B^j$ (body frames) and a world reference frame $\mathcal{F}_W$, as shown in Fig. 1.5.

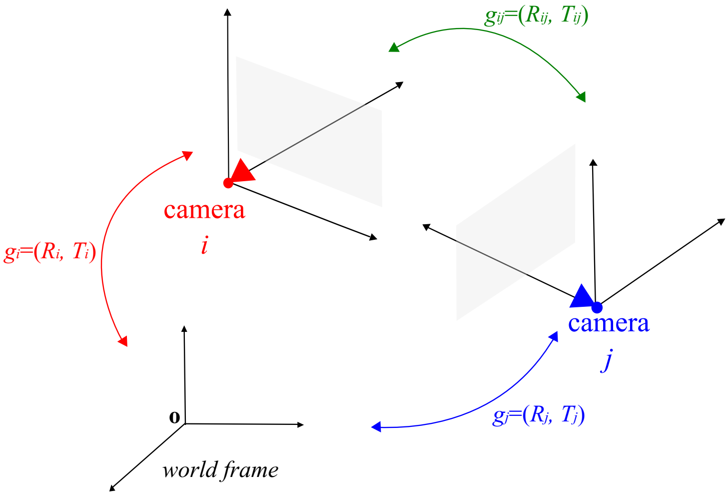


Figure 1.5: Example of relative body frames in a 3D camera network.

The absolute poses of frames i and j (i.e. $\mathcal{F}_B^i$, $\mathcal{F}_B^j$) are given by $g_i = (\mathbf{R}_{Oi}, \mathbf{T}_{Oi})$ and $g_j = (\mathbf{R}_{Oj}, \mathbf{T}_{Oj})$ respectively. How can we define and compute the relative pose $g_{ij} = (\mathbf{R}_{ij}, \mathbf{T}_{ij})$? Recall that $\mathbf{R}_{Oi}$ is the rotation of frame i with respect to the world, that is a "Body2World" rotation, measuring how the body frame $\mathcal{F}_B^i$ is rotated with respect to $\mathcal{F}_W$.

Typical framework for mobile agents:
Taking a generic 3D vector $\mathbf{v}_B$ in the body frame, Body2World means:

$$\mathbf{v}_W = \mathbf{R}_{WB}\mathbf{v}_B \tag{1.34}$$

then, similarly, we have

$$\begin{aligned} O \to i: &\quad \mathbf{v}_O = \mathbf{R}_{Oi}\mathbf{v}_i + \mathbf{T}_{Oi} &(1.35)\\ O \to j: &\quad \mathbf{v}_O = \mathbf{R}_{Oj}\mathbf{v}_j + \mathbf{T}_{Oj} &(1.36)\\ i \to j: &\quad \mathbf{v}_i = \mathbf{R}_{ij}\mathbf{v}_j + \mathbf{T}_{ij} &(1.37)\end{aligned}$$

Moreover, (1.35) and (1.36) implies that

$$\begin{aligned} \mathbf{R}_{Oi}\mathbf{v}_i + \mathbf{T}_{Oi} &= \mathbf{R}_{Oj}\mathbf{v}_j + \mathbf{T}_{Oj} &(1.38)\\ \mathbf{v}_i + \mathbf{R}_{Oi}^\top\mathbf{T}_{Oi} &= \mathbf{R}_{Oi}^\top\mathbf{R}_{Oj}\mathbf{v}_j + \mathbf{R}_{Oi}^\top\mathbf{T}_{Oj} &(1.39)\end{aligned}$$

and, finally,

$$\mathbf{v}_i = \mathbf{R}_{Oi}^\top\mathbf{R}_{Oj}\mathbf{v}_j + \mathbf{R}_{Oi}^\top(\mathbf{T}_{Oj} - \mathbf{T}_{Oi}) \tag{1.40}$$

By comparing the two expressions in (1.46) and (1.49) we obtain

$$\begin{aligned} \mathbf{R}_{ij} &= \mathbf{R}_{Oi}^\top\mathbf{R}_{Oj} &(1.41)\\ \mathbf{T}_{ij} &= \mathbf{R}_{Oi}^\top(\mathbf{T}_{Oj} - \mathbf{T}_{Oi}) &(1.42)\end{aligned}$$

Typical framework for camera networks:

Conversely, taking a generic 3D vector $\mathbf{v}_W$ in the world frame, World2Body means:

$$\mathbf{v}_B = \mathbf{R}_{BW}\mathbf{v}_W. \tag{1.43}$$

If we consider 3D points as vectors we have

$$i \to O: \quad \mathbf{Q}_i = \mathbf{R}_{iO}\mathbf{Q}_O + \mathbf{T}_{iO} \tag{1.44}$$

$$j \to O: \quad \mathbf{Q}_j = \mathbf{R}_{jO}\mathbf{Q}_O + \mathbf{T}_{jO} \tag{1.45}$$

$$i \to j: \quad \mathbf{Q}_i = \mathbf{R}_{ij}\mathbf{Q}_j + \mathbf{T}_{ij} \tag{1.46}$$

Moreover, (1.45) implies that

$$\mathbf{Q}_O = \mathbf{R}_{jO}^\top(\mathbf{Q}_j - \mathbf{T}_{jO}) \tag{1.47}$$

and replacing in (1.44)

$$\mathbf{Q}_i = \mathbf{R}_{iO}\mathbf{R}_{jO}^\top(\mathbf{Q}_j - \mathbf{T}_{jO}) + \mathbf{T}_{iO} \tag{1.48}$$

and, finally,

$$\mathbf{Q}_i = \mathbf{R}_{iO}\mathbf{R}_{jO}^\top\mathbf{Q}_j + \mathbf{T}_{iO} - \mathbf{R}_{iO}\mathbf{R}_{jO}^\top\mathbf{T}_{jO} \tag{1.49}$$

By comparing the two expressions in (1.46) and (1.49) we obtain

$$\mathbf{R}_{ij} = \mathbf{R}_{iO}\mathbf{R}_{jO}^\top \tag{1.50}$$

$$\mathbf{T}_{ij} = \mathbf{T}_{iO} - \mathbf{R}_{iO}\mathbf{R}_{jO}^\top\mathbf{T}_{jO} \tag{1.51}$$

With these concepts and if we think of a network of many agents, we can state the following

Definition 6 (Oriented network) *A network of agents is said to be oriented if there is a set of relative orientation* $\{\mathbf{R}_{ij} \in \mathbb{SO}(3)\}$ *such that the absolute orientation of all agents* $\{\mathbf{R}_{Oi} \in \mathbb{SO}(3)\}$ *is uniquely determined once the frame of any agent is absolutely fixed.*

Remark 5 *Two rotation matrices* $\mathbf{R}(\alpha)$ *and* $\mathbf{R}(\beta)$ *can commute in the multiplication operation, i.e.,* $\mathbf{R}(\alpha)\mathbf{R}(\beta) = \mathbf{R}(\beta)\mathbf{R}(\alpha)$, *only if they belong to* $\mathbb{SO}(2)$, *which is an Abelian (commutative) group.*
Conversely, $\mathbb{SO}(3)$ *is not an Abelian group!*

The geometry of $\mathbb{SO}(3)$ and $so(3)$

Consider a generic 3D vector $\boldsymbol{\omega} = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^\top$. The skew-symmetric operator associated to the vector ω is defined as

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{1.52}$$

We also have the following two definitions:

Definition 7 (Commutator) *The commutator of two square matrices $\mathbf{A}$ and $\mathbf{B}$ is given by $[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA}$.*

Note that if two matrices commute, then the commutator is equal to zero.

Definition 8 (Lie algebra) *A linear Lie algebra is a vector space of matrices that is closed under the commutator operator, meaning that the commutator of any two matrices in the set is also in the set.*

We can now state:

Theorem 1 ($so(3)$) *$so(3)$, the set of all skew symmetric 3×3 real matrices, is a Lie algebra.*

Proof: If $\mathbf{A} \in so(3)$ and $\mathbf{B} \in so(3)$, we have that $\mathbf{A}^\top = -\mathbf{A}$ and $\mathbf{B}^\top = -\mathbf{B}$. The transposed commutator results

$$[\mathbf{A}, \mathbf{B}]^\top = (\mathbf{AB} - \mathbf{BA})^\top = \mathbf{B}^\top\mathbf{A}^\top - \mathbf{A}^\top\mathbf{B}^\top = \mathbf{BA} - \mathbf{AB} = -[\mathbf{A}, \mathbf{B}], \tag{1.53}$$

hence also the results is in $so(3)$ and the set is closed under the commutator. □

It results that a link between $\mathbb{SO}(3)$ and $so(3)$ is yielded by the logarithmic and exponential maps[1] as stated formally by the two theorems that follow.

Theorem 2 (Exponential map) *If $\mathbf{Q} \in so(3)$, then $e^{\mathbf{Q}} \in \mathbb{SO}(3)$.*

Proof: Recalling that $\mathbb{SO}(3)$ is given by

$$\mathbb{SO}(3) = \left\{\mathbf{R} \in \mathbb{R}^{3\times3} : \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \quad \det(\mathbf{R}) = 1\right\}, \tag{1.56}$$

and that $so(3)$ is the set of skew symmetric matrices, we need to prove that

1. $\left(e^{\mathbf{Q}}\right)^\top e^{\mathbf{Q}} = \mathbf{I}$: by resorting to the Taylor expansion we have

$$\left(e^{\mathbf{Q}}\right)^\top e^{\mathbf{Q}} = \left(\mathbf{I} + \mathbf{Q} + \frac{\mathbf{Q}^2}{2!} + \frac{\mathbf{Q}^3}{3!} + \dots\right)^\top \left(\mathbf{I} + \mathbf{Q} + \frac{\mathbf{Q}^2}{2!} + \frac{\mathbf{Q}^3}{3!} + \dots\right) \tag{1.57}$$

$$= \left(\mathbf{I} - \mathbf{Q} + \frac{\mathbf{Q}^2}{2!} - \frac{\mathbf{Q}^3}{3!} + \dots\right) \left(\mathbf{I} + \mathbf{Q} + \frac{\mathbf{Q}^2}{2!} + \frac{\mathbf{Q}^3}{3!} + \dots\right) \tag{1.58}$$

$$= e^{-\mathbf{Q}}e^{\mathbf{Q}} = \mathbf{I} \tag{1.59}$$

[1] In general, these two can be computed via Taylor expansion. Let $\mathbf{A}$ be any matrix, then

$$\exp(\mathbf{A}) = \mathbf{I} + \mathbf{A} + \frac{\mathbf{A}^2}{2!} + \frac{\mathbf{A}^3}{3!} + \dots \tag{1.54}$$

$$\log(\mathbf{A}) = (\mathbf{A} - \mathbf{I}) - \frac{(\mathbf{A} - \mathbf{I})^2}{2} + \frac{(\mathbf{A} - \mathbf{I})^3}{3} - \dots \tag{1.55}$$

2. $|e^{\mathbf{Q}}| = 1$: by noting that skew symmetric matrices have one eigenvalue λ_0 equal to zero e two eigenvalues λ_{1-2} that are pure imaginary complex conjugate numbers, we have that

$$|e^{\mathbf{Q}}| = e^{\lambda_0} e^{\lambda_1} e^{\lambda_2} = e^{\lambda_0+\lambda_1+\lambda_2} = e^0 = 1. \tag{1.60}$$

□

Remark 6 *: In a more compact fashion, given an $\boldsymbol{\omega} \in \mathbb{R}^3$ and $[\boldsymbol{\omega}]_\times \in so(3)$ the Rodrigues formula holds:*

$$\begin{cases} [\boldsymbol{\omega}]_\times \in so(3) \\ \theta^2 = \boldsymbol{\omega}^\top \boldsymbol{\omega} \end{cases} \implies \mathbf{R} = \exp([\boldsymbol{\omega}]_\times) = \mathbf{I} + \frac{\sin\theta}{\theta}[\boldsymbol{\omega}]_\times + \frac{1-\cos\theta}{\theta^2}[\boldsymbol{\omega}]_\times^2 \tag{1.61}$$

Note that this is the sum of symmetric and skew-symmetric terms.
The final equation obtained in (1.61) expresses the rotation of an angle θ in radians around the axis given by $\boldsymbol{\omega}$ (according to the axis-angle rotation representation*).*

Conversely, we have that:

Theorem 3 (Logarithmic map) *If $\mathbf{R} \in \mathbb{SO}(3)$, then $log(\mathbf{R}) \in so(3)$.*

Remark 7 *: Even in this case, it is possible to adopt a more compact formulation by using the following formulas:*

$$\begin{cases} \mathbf{R} \in \mathbb{SO}(3) \\ \theta = \arccos\dfrac{\mathrm{tr}(\mathbf{R})-1}{2} \end{cases} \implies [\boldsymbol{\omega}]_\times = \log(\mathbf{R}) = \frac{\theta}{2\sin\theta}(\mathbf{R}-\mathbf{R}^\top) \tag{1.62}$$

To prove (1.62) we start from (1.61) by considering the symmetric and the skew-symmetric terms separately:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \frac{\sin\theta}{\theta}\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} + \frac{1-\cos\theta}{\theta^2}\begin{bmatrix} -\omega_y^2-\omega_z^2 & \omega_x\omega_y & \omega_x\omega_z \\ \omega_y\omega_x & -\omega_x^2-\omega_z^2 & \omega_y\omega_z \\ \omega_z\omega_x & \omega_z\omega_y & -\omega_x^2-\omega_y^2 \end{bmatrix}$$

- it can be derived from the skew-symmetric:

$$\begin{cases} r_{32}-r_{23} = 2\dfrac{\sin\theta}{\theta}\omega_x \\ r_{13}-r_{31} = 2\dfrac{\sin\theta}{\theta}\omega_y \\ r_{21}-r_{12} = 2\dfrac{\sin\theta}{\theta}\omega_z \end{cases} \Rightarrow \begin{matrix} \omega_x = \dfrac{\theta}{2\sin\theta}(r_{32}-r_{23}) \\ \omega_y = \dfrac{\theta}{2\sin\theta}(r_{13}-r_{31}) \\ \omega_z = \dfrac{\theta}{2\sin\theta}(r_{21}-r_{12}) \end{matrix} \tag{1.63}$$

which can be written in matricial form as

$$\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \frac{\theta}{2\sin\theta}\begin{bmatrix} 0 & -r_{21}+r_{12} & r_{13}-r_{31} \\ r_{21}-r_{12} & 0 & -r_{32}+r_{23} \\ -r_{13}+r_{31} & r_{32}-r_{23} & 0 \end{bmatrix} = \frac{\theta}{2\sin\theta}\left(\mathbf{R}-\mathbf{R}^\top\right) \tag{1.64}$$

- the symmetric terms can be used to compute the trace of $\mathbf{R}$ as

$$\mathrm{tr}(\mathbf{R}) = r_{11}+r_{22}+r_{33} = 3 + \frac{1-\cos\theta}{\theta^2}2\left(\omega_x^2+\omega_y^2+\omega_z^2\right) = 1+2\cos\theta \tag{1.65}$$

which can be inverted to provide $\theta = \arccos\dfrac{\mathrm{tr}(\mathbf{R})-1}{2}$.

We conclude the study of the relations between $so(3)$ and $\mathbb{SO}(3)$ by stating the important result:

Theorem 4 (Tangent space to $\mathbb{SO}(3)$) *The tangent space to $\mathbb{SO}(3)$ at a point $\mathbf{R}$ (see Fig. 1.6) is identified as*

$$\mathbf{T}_R(\mathbb{SO}(3)) = \left\{\mathbf{R}[\mathbf{q}]_\times : [\mathbf{q}]_\times \in so(3), \forall \mathbf{q} \in \mathbb{R}^3\right\} \tag{1.66}$$

where the Lie Algebra $so(3)$ has to be intended as the set of 3 skew-symmetric matrices.

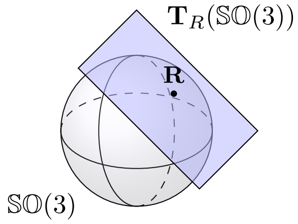


Figure 1.6: Representation of the tangent space at a point $\mathbf{R}$ of $\mathbb{SO}(3)$.

Proof: We proceed in two steps: first, we show that $so(3)$ is tangent to $\mathbb{SO}(3)$ at the identity $\mathbf{R} = \mathbf{I}$; then we generalize for a generic $\mathbf{R} \in \mathbb{SO}(3)$.

1. The proof of this fact lays on showing that the generators of $so(3)$ are the derivatives of the rotations around each axis, evaluated at the identity:

$$\frac{\partial \mathbf{R}_x(\alpha)}{\partial \alpha} = \left[\begin{array}{c|cc} 0 & 0 & 0 \\ \hline 0 & -\mathrm{s}_\alpha & -\mathrm{c}_\alpha \\ 0 & \mathrm{c}_\alpha & -\mathrm{s}_\alpha \end{array}\right] \qquad \frac{\partial \mathbf{R}_y(\beta)}{\partial \beta} = \begin{bmatrix} -\mathrm{s}_\beta & 0 & \mathrm{c}_\beta \\ 0 & 0 & 0 \\ -\mathrm{c}_\beta & 0 & \mathrm{s}_\beta \end{bmatrix} \qquad \frac{\partial \mathbf{R}_z(\gamma)}{\partial \gamma} = \left[\begin{array}{cc|c} -\mathrm{s}_\gamma & -\mathrm{c}_\gamma & 0 \\ \mathrm{c}_\gamma & -\mathrm{s}_\gamma & 0 \\ \hline 0 & 0 & 0 \end{array}\right] \tag{1.67}$$

$$\Downarrow \alpha = 0 \qquad\qquad \Downarrow \beta = 0 \qquad\qquad \Downarrow \gamma = 0$$

$$\mathbf{G}_x = \left[\begin{array}{c|cc} 0 & 0 & 0 \\ \hline 0 & 0 & -1 \\ 0 & 1 & 0 \end{array}\right] \qquad \mathbf{G}_y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \qquad \mathbf{G}_z = \left[\begin{array}{cc|c} 0 & -1 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right] \tag{1.68}$$

   In this way, an element of $so(3)$ can be computed as a linear combination given the basis $\{\mathbf{G}_x, \mathbf{G}_y, \mathbf{G}_z\}$:

$$a\mathbf{G}_x + b\mathbf{G}_y + c\mathbf{G}_z = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} \in so(3) \tag{1.69}$$

   Hence, we have the fundamental relation

$$\mathbf{T}_I(\mathbb{SO}(3)) = \left\{[\mathbf{q}]_\times : [\mathbf{q}]_\times \in so(3), \forall \mathbf{q} \in \mathbb{R}^3\right\} \tag{1.70}$$

   equivalent to state that the Lie Algebra $so(3)$ is the tangent space at the identity element of $\mathbb{SO}(3)$.

2. Now, to obtain the tangent space at any other point $\mathbf{R}$, as $\mathbf{R}$ is rotated w.r.t. $\mathbf{I}$ we need to rotate $so(3)$ by $\mathbf{R}$. Therefore, an element of the tangent space $\mathbf{T}_R(\mathbb{SO}(3))$ can be

seen as the composition of the rotation matrix $\mathbf{R}$ applied on the skew-symmetric matrix associated to any vector in $\mathbb{R}^3$.

We move from the origin ($\mathbf{I}$) to $\mathbf{R}$ in order to describe properly $\mathbf{T}_R(\mathbb{SO}(3))$ (Fig. 1.7):

$$\mathbf{T}_R(\mathbb{SO}(3)) = \left\{\mathbf{R}[\mathbf{q}]_\times : [\mathbf{q}]_\times \in so(3), \forall \mathbf{q} \in \mathbb{R}^3\right\} \tag{1.71}$$

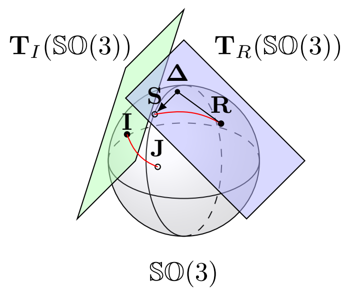


Figure 1.7: Changing origin from $\mathbf{I}$ to $\mathbf{R}$.

$\square$

Remark 8 *: The scenario for $\mathbb{SO}(2)$ and the connected algebra $so(2)$ work similarly to what just explained for the general $\mathbb{SO}(3) - so(3)$. In summary:*

- *given a 2D rotation angle θ we have*

$$[\theta]_\times = \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} \in so(2) \tag{1.72}$$

- *there is only one generator*

$$\frac{\partial \mathbf{R}(\theta)}{\partial \theta} = \begin{bmatrix} -\mathrm{s}_\theta & -\mathrm{c}_\theta \\ \mathrm{c}_\theta & \mathrm{s}_\theta \end{bmatrix} \quad \stackrel{\theta=0}{\Longrightarrow} \quad \mathbf{G} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{1.73}$$

- *Rodrigues formula still holds:*

$$\mathbf{R} = \exp([\boldsymbol{\theta}]_\times) = \mathbf{I} + \frac{\sin\theta}{\theta}[\boldsymbol{\theta}]_\times + \frac{1-\cos\theta}{\theta^2}[\boldsymbol{\theta}]^2_\times \tag{1.74}$$

whence it follows (by considering the symmetric part) that $\theta = \arccos\dfrac{\mathrm{tr}(\mathbf{R})}{2}$.

In practice (Fig. 1.8): $so(3) \xrightarrow{\exp} \mathbb{SO}(3)$ and viceversa $\mathbb{SO}(3) \xrightarrow{\log} so(3)$.

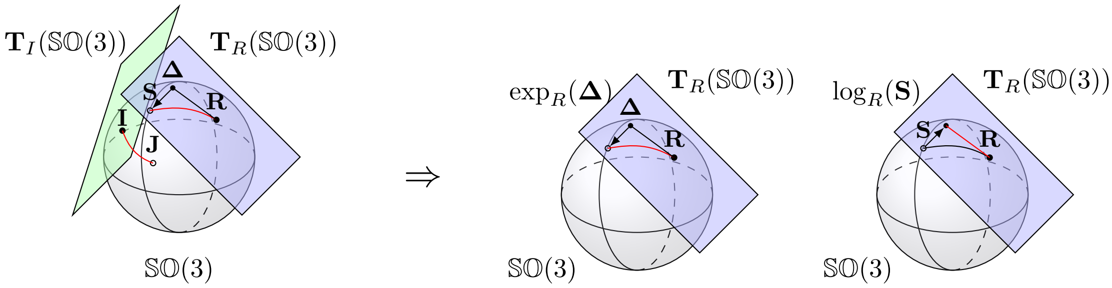


Figure 1.8: Exponential and logarithm maps acting on $\mathbb{SO}(3)$ w.r.t. the rotation matrix $\mathbf{R}$.

In summary: we have $\mathbf{S}$ and $\mathbf{R}$ in $\mathbb{SO}(3)$ and $\mathbf{\Delta}$ is the projection of $\mathbf{S}$ onto $\mathbf{T}_R(\mathbb{SO}(3))$,

$$\exists \mathbf{J} \in \mathbb{SO}(3) : \mathbf{S} = \mathbf{RJ} \in \mathbb{SO}(3) \quad \Rightarrow \quad \mathbf{J} = \mathbf{R}^\top \mathbf{S} \;\;\&\;\; \mathbf{I} = \mathbf{R}^\top \mathbf{R} \tag{1.75}$$

- $\mathbf{R} \& \mathbf{\Delta} \Rightarrow S \rightsquigarrow \mathbf{T}_R(\mathbb{SO}(3) \curvearrowright so(3) \curvearrowright \mathbb{SO}(3) \curvearrowright \mathbb{SO}(3)$

$$\exists [\boldsymbol{\omega}]_\times \in so(3) : e^{[\boldsymbol{\omega}]_\times} = \mathbf{J} \quad \Rightarrow \quad \mathbf{S} = \mathbf{R} e^{[\boldsymbol{\omega}]_\times} \tag{1.76}$$

$$\mathbf{\Delta} \in \mathbf{T}_R(\mathbb{SO}(3)) : \mathbf{\Delta} = \mathbf{R}[\boldsymbol{\omega}]_\times \quad \Rightarrow \quad [\boldsymbol{\omega}]_\times = \mathbf{R}^\top \mathbf{\Delta} \tag{1.77}$$

$$\mathbf{S} = \mathbf{R} e^{\mathbf{R}^\top \mathbf{\Delta}} \tag{1.78}$$

- $\mathbf{R} \& \mathbf{S} \Rightarrow \mathbf{\Delta} \rightsquigarrow \mathbb{SO}(3) \curvearrowright \mathbb{SO}(3) \curvearrowright so(3) \curvearrowright \mathbf{T}_R(\mathbb{SO}(3))$

$$\exists [\boldsymbol{\omega}]_\times \in so(3) : [\boldsymbol{\omega}]_\times = \log \mathbf{J} \quad \Rightarrow \quad [\boldsymbol{\omega}]_\times = \log \mathbf{R}^\top \mathbf{S} \tag{1.79}$$

$$\mathbf{\Delta} \in \mathbf{T}_R(\mathbb{SO}(3)) : \mathbf{\Delta} = \mathbf{R}[\boldsymbol{\omega}]_\times \quad \Rightarrow \quad \mathbf{\Delta} = \mathbf{R}[\boldsymbol{\omega}]_\times \tag{1.80}$$

$$\mathbf{\Delta} = \mathbf{R} \log \mathbf{R}^\top \mathbf{S} \tag{1.81}$$

Remark 9 *In summary:*

- *the exponential map is the diffeomorphism that associates to each point $\mathbf{\Delta}$ on the tangent plane of $\mathbf{R}$ (in the neighborhood) a point $\mathbf{S}$ on the unique geodesic passing through $\mathbf{R}$ in the direction of $\mathbf{\Delta}$;*
- *the logarithmic map is the diffeomorphism that associates to each point $\mathbf{S}$ on $\mathbb{SO}(3)$ a point $\mathbf{\Delta}$ on the tangent plane of $\mathbf{R}$ (in the neighborhood) representing the direction of the unique geodesic passing through $\mathbf{R}$ in the direction of $\mathbf{S}$.*

Distance on $\mathbb{SO}(3)$

With all these elements we can now tackle the problem of metrics. In other words, how can we measure the distance between two rotations? In the following we will consider the chordal metrics (distance is measured along chords in $\mathbb{SO}(3)$) and the geodesic metrics (distance is measured along arcs on $\mathbb{SO}(3)$):

1. Frobenius[2] or chordal metrics:

$$d_F(\mathbf{R},\mathbf{S}) = \sqrt{\frac{1}{2}\|\mathbf{R}-\mathbf{S}\|_F^2} \tag{1.83}$$

Example 1 (Frobenius distance on $\mathbb{SO}(2)$) *If we consider two planar rotations of angles α and β respectively*

$$\mathbf{R} = \mathbf{R}_\alpha = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \qquad \mathbf{S} = \mathbf{R}_\beta = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \tag{1.84}$$

the Frobenius distance between the two results

$$d_F(\mathbf{R}_\alpha,\mathbf{R}_\beta) = \sqrt{\frac{1}{2}\|\mathbf{R}_\alpha-\mathbf{R}_\beta\|_F^2} = \cdots = 2\sin\frac{\alpha-\beta}{2} \tag{1.85}$$

Recalling the Chord theorem from trigonometry, this is the length of a chord in the goniometric circumference with angle at the center equal to $(\alpha-\beta)$.

2. Riemannian geodesic metrics:

$$d_R(\mathbf{R},\mathbf{S}) = \sqrt{\frac{1}{2}\|\log(\mathbf{S}^\top\mathbf{R})\|_F^2} \tag{1.86}$$

Example 2 (Riemannian distance on $\mathbb{SO}(2)$) *If we consider two planar rotations of angles α and β as in (1.84), the Riemannian distance between the two results*

$$d_R(\mathbf{R}_\alpha,\mathbf{R}_\beta) = \sqrt{\frac{1}{2}\left\|\log(\mathbf{R}_\beta^\top\mathbf{R}_\alpha)\right\|_F^2} = \cdots = \alpha-\beta \tag{1.87}$$

This is the length of arc in the goniometric circumference with angle at the center equal to $(\alpha-\beta)$ (measured in radians).

Please note that there are also other metrics such as:

3. Deviation from the identity (equivalent to chordal metrics):

$$d_I(\mathbf{R},\mathbf{S}) = \sqrt{\frac{1}{2}\|\mathbf{S}^\top\mathbf{R}-\mathbf{I}\|_F^2} \tag{1.88}$$

4. Hyperbolic metric:

$$d_H(\mathbf{R},\mathbf{S}) = \sqrt{\frac{1}{2}\|\log(\mathbf{R})-\log(\mathbf{S})\|_F^2} \tag{1.89}$$

[2]Recalling that

$$\|\mathbf{A}\|_F^2 = \sum\sum|a_{ij}|^2 = \mathrm{tr}(\mathbf{A}^\top\mathbf{A}) = \sum\sigma_A^2. \tag{1.82}$$

Rotation matrix derivative

To compute the rotation matrix time derivative $\dot{\mathbf{R}}$, we start from the relation

$$\mathbf{R}\mathbf{R}^\top = \mathbf{I} \tag{1.90}$$

and take the derivatives on both sides

$$\dot{\mathbf{R}}\mathbf{R}^\top + \mathbf{R}\dot{\mathbf{R}}^\top = \mathbf{0} \Rightarrow \dot{\mathbf{R}}\mathbf{R}^\top = -\mathbf{R}\dot{\mathbf{R}}^\top = -\left(\dot{\mathbf{R}}\mathbf{R}^\top\right)^\top \tag{1.91}$$

It follows that $\dot{\mathbf{R}}\mathbf{R}^\top$ is equal to a skew-symmetric matrix $\mathbf{S}$, hence

$$\dot{\mathbf{R}}\mathbf{R}^\top = \mathbf{S} \Rightarrow \dot{\mathbf{R}} = \mathbf{S}\mathbf{R}. \tag{1.92}$$

To understand the meaning of the matrix $\mathbf{S}$ we proceed as follows.
We consider the two frames $\mathcal{F}_B$ and $\mathcal{F}_W$ rotated by $\mathbf{R}_{WB}$ and a point $\mathbf{p}_B \in \mathcal{F}_B$

$$\mathbf{p}_W = \mathbf{R}_{WB}\mathbf{p}_B \tag{1.93}$$

and a rotation of $\mathcal{F}_B$ with angular velocity $\boldsymbol{\omega}_W$, which is a vector in $\mathbb{R}^3$

$$\boldsymbol{\omega}_W = \mathbf{R}_{WB}\boldsymbol{\omega}_B \tag{1.94}$$

The derivative of (1.93) yields

$$\dot{\mathbf{p}}_W = \dot{\mathbf{R}}_{WB}\mathbf{p}_B + \mathbf{R}_{WB}\dot{\mathbf{p}}_B = \dot{\mathbf{R}}_{WB}\mathbf{p}_B \tag{1.95}$$

since the point is fixed in $\mathcal{F}_B$. By recalling the relation between angular ($\boldsymbol{\omega}_W$) and linear velocity ($\dot{\mathbf{p}}_W$) we have

$$\dot{\mathbf{p}}_W = \boldsymbol{\omega}_W \times \mathbf{p}_W = [\boldsymbol{\omega}_W]_\times \mathbf{p}_W = [\boldsymbol{\omega}_W]_\times \mathbf{R}_{WB}\mathbf{p}_B \tag{1.96}$$

and by equalizing these last two equations we get

$$\dot{\mathbf{R}}_{WB}\mathbf{p}_B = [\boldsymbol{\omega}_W]_\times \mathbf{R}_{WB}\mathbf{p}_B \tag{1.97}$$

which is true for any point $\mathbf{p}_B \in \mathcal{F}_B$; hence, it follows

$$\dot{\mathbf{R}}_{WB} = [\boldsymbol{\omega}_W]_\times \mathbf{R}_{WB}. \tag{1.98}$$

By taking the transpose, we get

$$\dot{\mathbf{R}}_{BW} = -\mathbf{R}_{BW}[\boldsymbol{\omega}_W]_\times. \tag{1.99}$$

By recalling the skew symmetric operator property

$$[\mathbf{R}\boldsymbol{\omega}]_\times = \mathbf{R}[\boldsymbol{\omega}]_\times \mathbf{R}^\top \tag{1.100}$$

we obtain also, by substitution,

$$[\boldsymbol{\omega}_W]_\times = [\mathbf{R}_{WB}\boldsymbol{\omega}_B]_\times = \mathbf{R}_{WB}[\boldsymbol{\omega}_B]_\times \mathbf{R}_{BW}. \tag{1.101}$$

It results:

$$\dot{\mathbf{R}}_{WB} = \mathbf{R}_{WB}[\boldsymbol{\omega}_B]_\times \tag{1.102}$$

and

$$\dot{\mathbf{R}}_{BW} = -[\boldsymbol{\omega}_B]_\times \mathbf{R}_{BW}. \tag{1.103}$$

1.2.2 Attitude representation: quaternions

Definition

The *quaternion* is an hyper-complex entity that represents the extension of a complex number in a higher dimensional space:

$$\mathbf{q} = \underbrace{\eta}_{real} + \underbrace{i\epsilon_i + j\epsilon_j + k\epsilon_k}_{complex} = \eta + \boldsymbol{\epsilon} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \tag{1.104}$$

where (i, j, k) is a coordinate frame abiding the Hamilton's rules:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{1.105}$$

$$ij = k \quad jk = i \quad ki = j \tag{1.106}$$

$$ji = -k \quad kj = -i \quad ik = -j \tag{1.107}$$

We can provide a physical interpretation to the quaternion if we consider a rotation (given by η) around a certain axis (given by $\boldsymbol{\epsilon}$) and we assign

$$\mathbf{q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \mathbf{e}\sin\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{1.108}$$

with $\mathbf{e}$ identifying a unit vector: $\mathbf{q}$ is the rotation of θ around $\mathbf{e}$ and the quaternion becomes a *unit quaternion*. Indeed, it follows the norm computation:

$$\|\mathbf{q}\|^2 = \cos^2\left(\frac{\theta}{2}\right) + \sin^2\left(\frac{\theta}{2}\right) = \eta^2 + \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon} = 1 \tag{1.109}$$

Finally, it can also be observed that $\mathbf{q} \in \mathbb{S}^3$.

Quaternion and rotation matrix

Through the Rodrigues formula (1.61) and trigonometric half-angle formulas, the unit quaternion can be put in relation with the rotation matrix as follows

$$\mathbf{R} = \mathbf{I} + \frac{\sin\theta}{\theta}[\boldsymbol{\omega}]_\times + \frac{1-\cos\theta}{\theta^2}[\boldsymbol{\omega}]_\times^2 \quad \Longrightarrow \quad \mathbf{R}(\mathbf{q}) = \mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + 2[\boldsymbol{\epsilon}]_\times^2 \tag{1.110}$$

by recalling that

- from the unit quaternion definition

$$\eta = \cos\left(\frac{\theta}{2}\right) \qquad \boldsymbol{\epsilon} = \mathbf{e}\sin\left(\frac{\theta}{2}\right) \tag{1.111}$$

- $\boldsymbol{\omega}$ is the non normalized version of the rotation vector $\boldsymbol{e}$ and the norm of $\boldsymbol{\omega}$ is θ

$$\boldsymbol{e} = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \quad \Longrightarrow \quad [\boldsymbol{e}]_\times = \frac{[\boldsymbol{\omega}]_\times}{\|\boldsymbol{\omega}\|} \tag{1.112}$$

- $[\boldsymbol{\epsilon}]_\times$ is the skew-symmetric matrix associated to $\boldsymbol{\epsilon}$ and its square $[\boldsymbol{\epsilon}]^2_\times$ results to be symmetric

$$[\boldsymbol{\epsilon}]_\times = \begin{bmatrix} 0 & -\epsilon_k & \epsilon_j \\ \epsilon_k & 0 & -\epsilon_i \\ -\epsilon_j & \epsilon_i & 0 \end{bmatrix} \qquad [\boldsymbol{\epsilon}]^2_\times = \begin{bmatrix} -(\epsilon_j^2+\epsilon_k^2) & \epsilon_i\epsilon_j & \epsilon_i\epsilon_k \\ \epsilon_i\epsilon_j & -(\epsilon_i^2+\epsilon_k^2) & \epsilon_j\epsilon_k \\ \epsilon_i\epsilon_k & \epsilon_j\epsilon_k & -(\epsilon_i^2+\epsilon_j^2) \end{bmatrix} \tag{1.113}$$

From (1.110) we obtain, in a less compact form:

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} \eta^2+\epsilon_i^2-\epsilon_j^2-\epsilon_k^2 & 2\epsilon_i\epsilon_j - 2\eta\epsilon_k & 2\epsilon_i\epsilon_k + 2\eta\epsilon_j \\ 2\epsilon_i\epsilon_j + 2\eta\epsilon_k & \eta^2-\epsilon_i^2+\epsilon_j^2-\epsilon_k^2 & 2\epsilon_j\epsilon_k - 2\eta\epsilon_i \\ 2\epsilon_i\epsilon_k - 2\eta\epsilon_j & 2\epsilon_j\epsilon_k + 2\eta\epsilon_i & \eta^2-\epsilon_i^2-\epsilon_j^2+\epsilon_k^2 \end{bmatrix} \tag{1.114}$$

From (1.110) the fundamental property of quaternion double coverage can be seen: the same rotation $\mathbf{R}(\mathbf{q})$ can be represented by both $\mathbf{q}$ and $-\mathbf{q}$. Also, with respect to (1.108), it can be shown that $\mathbf{q}$ (or $-\mathbf{q}$) represents the rotation of θ around $\mathbf{e}$.

Quaternion conjugate and inverse

The quaternion conjugate $\bar{\mathbf{q}}$ is defined in agreement with the complex conjugate number as

$$\bar{\mathbf{q}} = \eta - i\epsilon_i - j\epsilon_j - k\epsilon_k = \eta - \boldsymbol{\epsilon} = \begin{bmatrix} \eta \\ -\boldsymbol{\epsilon} \end{bmatrix} \tag{1.115}$$

This allows to introduce the quaternion inverse (similarly to the standard complex number case)

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|^2} \tag{1.116}$$

that in the case of the unit quaternion results

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{1} = \bar{\mathbf{q}} \tag{1.117}$$

As in the case of the rotation matrix in $\mathbb{SO}(3)$, the inverse of the quaternion has the meaning of the inverse rotation (rotation around the same axis of the opposite angle)

$$\mathbf{R}(\mathbf{q}) \to \mathbf{R}(\mathbf{q})^{-1} = \mathbf{R}(\mathbf{q})^\top \quad \Leftrightarrow \quad \mathbf{q} \to \mathbf{q}^{-1} = \bar{\mathbf{q}} \tag{1.118}$$

Quaternion inner and outer products

Similarly to the vector space, we can define inner and outer products for quaternions, which result in a scalar and a vector respectively.
For the inner product we have:

$$\mathbf{q} = \mathbf{q}_1 \cdot \mathbf{q}_2 = (\eta_1 + \boldsymbol{\epsilon}_1)\cdot(\eta_2 + \boldsymbol{\epsilon}_2) = \eta_1\eta_2 + \boldsymbol{\epsilon}_1 \cdot \boldsymbol{\epsilon}_2 = \begin{bmatrix} \eta_1\eta_2 + \boldsymbol{\epsilon}_1^\top\boldsymbol{\epsilon}_2 \\ \mathbf{0} \end{bmatrix} \tag{1.119}$$

while the outer product results:

$$\mathbf{q} = \mathbf{q}_1 \times \mathbf{q}_2 = (\eta_1 + \boldsymbol{\epsilon}_1)\times(\eta_2 + \boldsymbol{\epsilon}_2) = \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2 = \begin{bmatrix} 0 \\ \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2 \end{bmatrix} \tag{1.120}$$

Quaternion combination (Hamilton product)

The rotation composition is obtained in the algebra of quaternions as with matrices

$$\mathbf{R}(\mathbf{q}_{tot}) = \mathbf{R}(\mathbf{q}_1)\mathbf{R}(\mathbf{q}_2) \quad \Leftrightarrow \quad \mathbf{q}_{tot} = \mathbf{q}_1 \circ \mathbf{q}_2 \tag{1.121}$$

Starting from

$$\mathbf{q}_1 = \eta_1 + i\epsilon_{1i} + j\epsilon_{1j} + k\epsilon_{1k} = \begin{bmatrix} \eta_1 \\ \boldsymbol{\epsilon}_1 \end{bmatrix} \qquad \mathbf{q}_2 = \eta_2 + i\epsilon_{2i} + j\epsilon_{2j} + k\epsilon_{2k} = \begin{bmatrix} \eta_2 \\ \boldsymbol{\epsilon}_2 \end{bmatrix} \tag{1.122}$$

the quaternion combination results

$$\begin{aligned} \mathbf{q}_{tot} &= \mathbf{q}_1 \circ \mathbf{q}_2 = (\eta_1 + i\epsilon_{1i} + j\epsilon_{1j} + k\epsilon_{1k}) \circ (\eta_2 + i\epsilon_{2i} + j\epsilon_{2j} + k\epsilon_{2k}) && (1.123) \\ &= (\eta_1\eta_2 - \epsilon_{1i}\epsilon_{2i} - \epsilon_{1j}\epsilon_{2j} - \epsilon_{1k}\epsilon_{2k}) + i\,(\eta_1\epsilon_{2i} + \eta_2\epsilon_{1i} + \epsilon_{1j}\epsilon_{2k} - \epsilon_{1k}\epsilon_{2j}) && (1.124) \\ &+ j\,(\eta_1\epsilon_{2j} + \eta_2\epsilon_{1j} + \epsilon_{1k}\epsilon_{2i} - \epsilon_{1i}\epsilon_{2k}) + k\,(\eta_1\epsilon_{2k} + \eta_2\epsilon_{1k} + \epsilon_{1i}\epsilon_{2j} - \epsilon_{1j}\epsilon_{2i}) && (1.125) \end{aligned}$$

that is equal to

$$\mathbf{q}_{tot} = \underbrace{\eta_1\eta_2 - \boldsymbol{\epsilon}_1^\top\boldsymbol{\epsilon}_2}_{\text{real part}} + \underbrace{\eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2}_{\text{hyper-complex part}} \tag{1.126}$$

(where we combine respectively $1+3$ terms for the real part and $3+3+6$ terms for the hyper-complex part). In compact form

$$\begin{aligned} \mathbf{q}_{tot} &= \mathbf{q}_1 \circ \mathbf{q}_2 = \begin{bmatrix} \eta_1 \\ \boldsymbol{\epsilon}_1 \end{bmatrix} \circ \begin{bmatrix} \eta_2 \\ \boldsymbol{\epsilon}_2 \end{bmatrix} && (1.127) \\ &= \begin{bmatrix} \eta_1\eta_2 - \boldsymbol{\epsilon}_1^\top\boldsymbol{\epsilon}_2 \\ \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2 \end{bmatrix} = \begin{bmatrix} \eta_2\eta_1 - \boldsymbol{\epsilon}_2^\top\boldsymbol{\epsilon}_1 \\ \eta_2\boldsymbol{\epsilon}_1 + \eta_1\boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_2 \times \boldsymbol{\epsilon}_1 \end{bmatrix} && (1.128) \\ &= \begin{bmatrix} \eta_1 & -\boldsymbol{\epsilon}_1^\top \\ \boldsymbol{\epsilon}_1 & \eta_1 I_3 + [\boldsymbol{\epsilon}_1]_\times \end{bmatrix} \begin{bmatrix} \eta_2 \\ \boldsymbol{\epsilon}_2 \end{bmatrix} = \begin{bmatrix} \eta_2 & -\boldsymbol{\epsilon}_2^\top \\ \boldsymbol{\epsilon}_2 & \eta_2 I_3 - [\boldsymbol{\epsilon}_2]_\times \end{bmatrix} \begin{bmatrix} \eta_1 \\ \boldsymbol{\epsilon}_1 \end{bmatrix} && (1.129) \\ &= M(\mathbf{q}_1)\mathbf{q}_2 = N(\mathbf{q}_2)\mathbf{q}_1 && (1.130) \end{aligned}$$

where

$$M(\mathbf{q}_1) = \begin{bmatrix} \eta_1 & -\boldsymbol{\epsilon}_1^\top \\ \boldsymbol{\epsilon}_1 & \eta_1 I_3 + [\boldsymbol{\epsilon}_1]_\times \end{bmatrix} \qquad N(\mathbf{q}_2) = \begin{bmatrix} \eta_2 & -\boldsymbol{\epsilon}_2^\top \\ \boldsymbol{\epsilon}_2 & \eta_2 I_3 - [\boldsymbol{\epsilon}_2]_\times \end{bmatrix} \tag{1.131}$$

Example 3 (Null rotation) *Note that if we combine a rotation with its inverse to obtain the null rotation we obtain*

$$\mathbf{q} \circ \bar{\mathbf{q}} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \circ \begin{bmatrix} \eta \\ -\boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \eta\eta - \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon} \\ \eta\boldsymbol{\epsilon} - \eta\boldsymbol{\epsilon} - \boldsymbol{\epsilon} \times \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} = \mathbf{q}_0 \tag{1.132}$$

which verifies the quaternion inverse definition.

Example 4 (Combination of planar rotations) *In the case of rotation around the same axis* $\mathbf{e}$

$$\mathbf{q}_1 = \underbrace{\cos\left(\frac{\theta_1}{2}\right)}_{\eta_1} + \underbrace{\mathbf{e}\sin\left(\frac{\theta_1}{2}\right)}_{\boldsymbol{\epsilon}_1} \qquad \mathbf{q}_2 = \underbrace{\cos\left(\frac{\theta_2}{2}\right)}_{\eta_2} + \underbrace{\mathbf{e}\sin\left(\frac{\theta_2}{2}\right)}_{\boldsymbol{\epsilon}_2} \tag{1.133}$$

we get from (1.126)

$$\begin{aligned}
\mathbf{q}_1 \circ \mathbf{q}_2 &= \eta_1\eta_2 - \boldsymbol{\epsilon}_1^\top \boldsymbol{\epsilon}_2 + \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2 && (1.134)\\
&= \cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) - \sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right)\mathbf{e}^\top\mathbf{e} + && (1.135)\\
&+ \cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right)\mathbf{e} + \cos\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right)\mathbf{e} + && (1.136)\\
&+ \sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right)\mathbf{e}\times\mathbf{e} && (1.137)\\
&= \left[\cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right) - \sin\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right)\right] + && (1.138)\\
&+ \left[\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) + \cos\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right)\right]\mathbf{e} && (1.139)\\
&= \left[\cos\left(\frac{\theta_1+\theta_2}{2}\right)\right] + \left[\sin\left(\frac{\theta_1+\theta_2}{2}\right)\right]\mathbf{e} && (1.140)
\end{aligned}$$

As expected we get a rotation around $\mathbf{e}$ *by an angle* $\theta_1 + \theta_2$.

Vector rotation

The rotation of a vector $\mathbf{v}$, namely $\mathbf{v}_{rotated} = \mathbf{R}(\mathbf{q})\mathbf{v}$ is obtained by writing vector $\mathbf{v}$ as a purely imaginary quaternion $\begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix}$ and computing the product

$$\begin{bmatrix} 0 \\ \mathbf{v}_{rotated} \end{bmatrix} = \mathbf{q} \circ \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} \circ \bar{\mathbf{q}} = \mathbf{M}(\mathbf{q}) \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} \circ \bar{\mathbf{q}} = \mathbf{M}(\mathbf{q})\mathbf{N}(\bar{\mathbf{q}}) \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} \tag{1.141}$$

It remains to check whether the multiplication by matrix $\mathbf{M}(\mathbf{q})\mathbf{N}(\bar{\mathbf{q}})$ allows to get a purely imaginary quaternion. By computing such a matrix it follows

$$\begin{aligned}
\mathbf{M}(\mathbf{q})\mathbf{N}(\bar{\mathbf{q}}) &= \begin{bmatrix} \eta & -\boldsymbol{\epsilon}^\top \\ \boldsymbol{\epsilon} & \eta_1\mathbf{I} + [\boldsymbol{\epsilon}]_\times \end{bmatrix} \begin{bmatrix} \eta & +\boldsymbol{\epsilon}^\top \\ -\boldsymbol{\epsilon} & \eta\mathbf{I} + [\boldsymbol{\epsilon}]_\times \end{bmatrix} && (1.142)\\
&= \begin{bmatrix} \eta^2 + \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon} & \eta\boldsymbol{\epsilon}^\top - \eta\boldsymbol{\epsilon}^\top - \boldsymbol{\epsilon}^\top[\boldsymbol{\epsilon}]_\times \\ \eta\boldsymbol{\epsilon} - \eta\boldsymbol{\epsilon} - [\boldsymbol{\epsilon}]_\times\boldsymbol{\epsilon} & \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top + \eta^2\mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + [\boldsymbol{\epsilon}]_\times^2 \end{bmatrix} && (1.143)\\
&= \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & * \end{bmatrix} && (1.144)
\end{aligned}$$

By multiplying this matrix by a pure imaginary quaternion, the result is again a pure imaginary quaternion, according to (1.141). Moreover, by working on the matricial entry $(2, 2)$ in (1.144) we obtain[3]:

$$\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top + \eta^2\mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + [\boldsymbol{\epsilon}]_\times^2 = \mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + 2[\boldsymbol{\epsilon}]_\times^2 \tag{1.149}$$

which is the Rodrigues-like formula for quaternions and rotation matrix $\mathbf{R}$. In other words,

$$\begin{bmatrix} 0 \\ \mathbf{v}_{rotated} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + 2[\boldsymbol{\epsilon}]_\times^2 \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{R}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{R}(\mathbf{q})\mathbf{v} \end{bmatrix} \tag{1.150}$$

As a mnemonic rule, we can observe (neglecting here vector/quaternion dimensions):

$$\mathbf{v}_i = \mathbf{R}_{ij}\mathbf{v}_j \quad \Longleftrightarrow \quad \mathbf{v}_i = \mathbf{q}_{ij} \circ \mathbf{v}_j \circ \mathbf{q}_{ji} = \mathbf{q}_{ij} \circ \mathbf{v}_j \circ \bar{\mathbf{q}}_{ij} \tag{1.151}$$

Distance on $\mathbb{S}^3$

We can compute distances with quaternions as with rotation matrices. In particular also in this case we will consider the chordal metrics and the geodesic metrics that in the quaternion algebra take the form:

1. Frobenius or chordal metric:

$$d_F^2(\mathbf{q}, \tilde{\mathbf{q}}) = \frac{1}{2}\|\mathbf{q} - \tilde{\mathbf{q}}\|_F^2 = \frac{1}{2}\|\mathbf{q} - \tilde{\mathbf{q}}\|_2^2 \tag{1.152}$$

By computing the norms in (1.152) [4], we obtain

$$\begin{aligned} d_F^2(\mathbf{q}, \tilde{\mathbf{q}}) &= \frac{1}{2}\left\{\begin{bmatrix} \eta - \tilde{\eta} \\ \boldsymbol{\epsilon} - \tilde{\boldsymbol{\epsilon}} \end{bmatrix}^\top \begin{bmatrix} \eta - \tilde{\eta} \\ \boldsymbol{\epsilon} - \tilde{\boldsymbol{\epsilon}} \end{bmatrix}\right\} && (1.153) \\ &= \frac{1}{2}\left\{(\eta - \tilde{\eta})^2 + (\boldsymbol{\epsilon} - \tilde{\boldsymbol{\epsilon}})^\top(\boldsymbol{\epsilon} - \tilde{\boldsymbol{\epsilon}})\right\} && (1.154) \\ &= \frac{1}{2}\left\{2 - 2(\eta\tilde{\eta} + \boldsymbol{\epsilon}^\top\tilde{\boldsymbol{\epsilon}})\right\} && (1.155) \\ &= 1 - \mathbf{q} \cdot \tilde{\mathbf{q}} && (1.156) \end{aligned}$$

Example 5 (Frobenius distance on $\mathbb{S}^3$) *If we consider two planar rotations of angles α and β respectively*

$$\mathbf{q}_\alpha = \begin{bmatrix} \cos\frac{\alpha}{2} \\ \mathbf{e}\sin\frac{\alpha}{2} \end{bmatrix} \qquad \mathbf{q}_\beta = \begin{bmatrix} \cos\frac{\beta}{2} \\ \mathbf{e}\sin\frac{\beta}{2} \end{bmatrix} \tag{1.157}$$

[3]The following equalities hold:

$$\begin{aligned} \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top + \eta^2\mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + [\boldsymbol{\epsilon}]_\times^2 &= \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top + (1 - \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon})\mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + [\boldsymbol{\epsilon}]_\times^2 && (1.145) \\ &= \begin{bmatrix} \epsilon_i^2 & \epsilon_i\epsilon_j & \epsilon_i\epsilon_k \\ \epsilon_i\epsilon_j & \epsilon_j^2 & \epsilon_j\epsilon_k \\ \epsilon_i\epsilon_k & \epsilon_j\epsilon_k & \epsilon_k^2 \end{bmatrix} + \mathbf{I} - (\epsilon_i^2 + \epsilon_j^2 + \epsilon_k^2)\mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + [\boldsymbol{\epsilon}]_\times^2 && (1.146) \\ &= \begin{bmatrix} -(\epsilon_j^2 + \epsilon_k^2) & \epsilon_i\epsilon_j & \epsilon_i\epsilon_k \\ \epsilon_i\epsilon_j & -(\epsilon_i^2 + \epsilon_k^2) & \epsilon_j\epsilon_k \\ \epsilon_i\epsilon_k & \epsilon_j\epsilon_k & -(\epsilon_i^2 + \epsilon_j^2) \end{bmatrix} + \mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + [\boldsymbol{\epsilon}]_\times^2 && (1.147) \\ &= \mathbf{I} + 2\eta[\boldsymbol{\epsilon}]_\times + 2[\boldsymbol{\epsilon}]_\times^2 && (1.148) \end{aligned}$$

[4]To be formal, given the fact that the quaternion has the double coverage property, $\|\mathbf{q} - \tilde{\mathbf{q}}\|_F = \min\{\|\mathbf{q} - \tilde{\mathbf{q}}\|, \|\mathbf{q} + \tilde{\mathbf{q}}\|\}$

the Frobenius distance between the two results

$$d_F(\mathbf{q}_\alpha, \mathbf{q}_\beta) = \sqrt{1 - \mathbf{q}_\alpha \cdot \mathbf{q}_\beta} = \cdots = \sqrt{1 - \cos\left(\frac{\alpha - \beta}{2}\right)} \tag{1.158}$$

This has not the same interpretation as the chordal length in the $\mathbb{SO}(3)$ case (see (1.83)), but it is in any case related to the angle through a projection *trigonometric function.*

2. Riemannian geodesic metric:

$$d_R(\mathbf{q}, \tilde{\mathbf{q}}) = 2\arccos(Real(\mathbf{q}^{-1} \circ \tilde{\mathbf{q}})) = 2\arccos(\mathbf{q} \cdot \tilde{\mathbf{q}}) \tag{1.159}$$

This formula can be obtained through the following steps:

$$d_R(\mathbf{q}, \tilde{\mathbf{q}}) = d_R(\mathbf{q}^{-1} \circ \mathbf{q}, \mathbf{q}^{-1} \circ \tilde{\mathbf{q}}) = d_R(\mathbf{q}_0, \mathbf{q}^{-1} \circ \tilde{\mathbf{q}}) \tag{1.160}$$

and observing that

$$\mathbf{q}^{-1} \circ \tilde{\mathbf{q}} \quad = \quad \begin{bmatrix} \eta\tilde{\eta} + \boldsymbol{\epsilon}^\top \tilde{\boldsymbol{\epsilon}} \\ \eta\tilde{\boldsymbol{\epsilon}} - \tilde{\eta}\boldsymbol{\epsilon} - \boldsymbol{\epsilon} \times \tilde{\boldsymbol{\epsilon}} \end{bmatrix} \tag{1.161}$$

where the $\arccos$ of the scalar part provides an arc-length.

Example 6 (Riemannian distance on $\mathbb{S}^3$) *If we consider two planar rotations of angles α and β as in (1.157), the Riemannian distance between the two results*

$$d_R(\mathbf{q}_\alpha, \mathbf{q}_\beta) = 2\arccos(\mathbf{q}_\alpha \cdot \mathbf{q}_\beta) = \cdots = \alpha - \beta \tag{1.162}$$

This is the length of arc in the goniometric circumference with angle at the center equal to $(\alpha - \beta)$ (measured in radians).

A summary is presented in the next figures, where the rotation distance is evaluated between 0 and 2π:

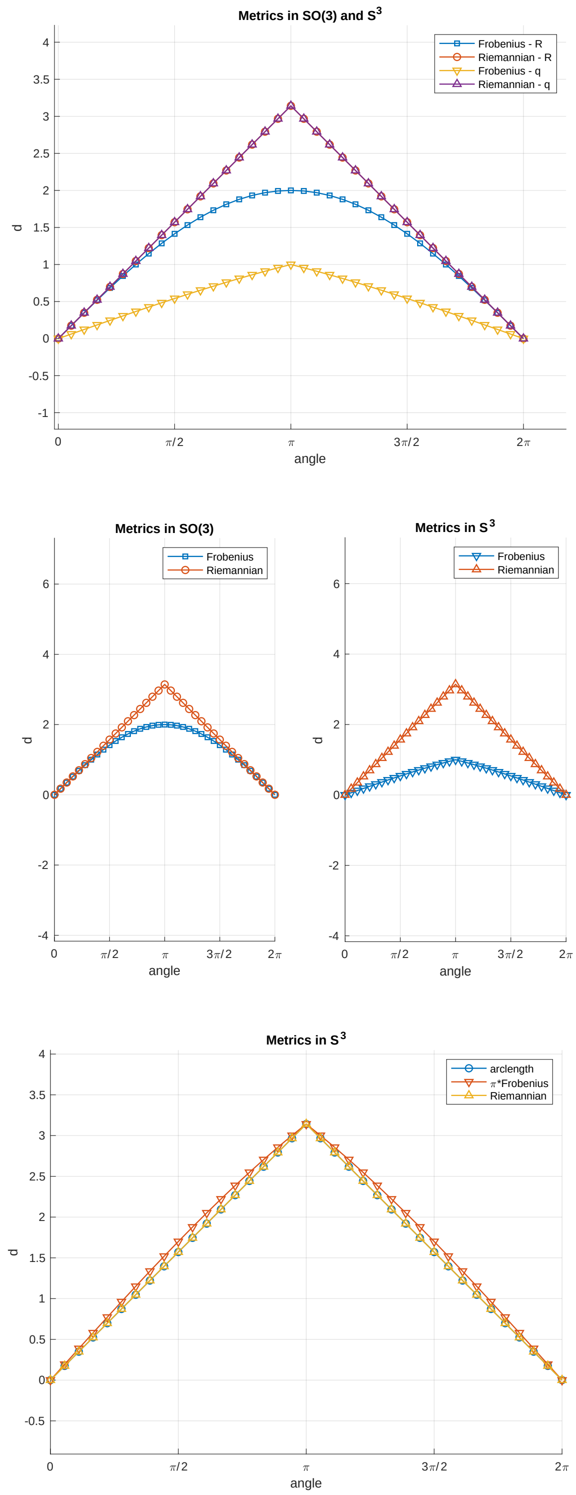


Figure 1.9: Rotation metrics.

Quaternion derivative

To compute the quaternion time derivative $\dot{\mathbf{q}}$, we start from the relation

$$\mathbf{q}_W(t+\Delta t) = \Delta\mathbf{q}_W \circ \mathbf{q}_W(t) \tag{1.163}$$

that relates the quaternion in $\mathcal{F}_W$ at time t with that at time $t+\Delta t$, with

$$\begin{aligned} \Delta\mathbf{q} &= \cos\frac{\Delta\theta}{2} + \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}\sin\frac{\Delta\theta}{2} && (1.164)\\ \Delta\theta &= \|\boldsymbol{\omega}\|\Delta t && (1.165) \end{aligned}$$

and we compute the incremental ratio. It follows:

$$\begin{aligned} \dot{\mathbf{q}}_W &= \lim_{\Delta t\to 0}\frac{\mathbf{q}_W(t+\Delta t)-\mathbf{q}_W(t)}{\Delta t} && (1.166)\\ &= \lim_{\Delta t\to 0}\frac{\Delta\mathbf{q}_W\circ\mathbf{q}_W(t)-\mathbf{q}_W(t)}{\Delta t} && (1.167)\\ &= \lim_{\Delta t\to 0}\frac{(\Delta\mathbf{q}_W-\mathbf{q}_0)\circ\mathbf{q}_W(t)}{\Delta t} && (1.168)\\ &= \lim_{\Delta t\to 0}\frac{\Delta\mathbf{q}_W-\mathbf{q}_0}{\Delta t}\circ\mathbf{q}_W(t). && (1.169) \end{aligned}$$

The limit term becomes

$$\begin{aligned} \lim_{\Delta t\to 0}\frac{\Delta\mathbf{q}_W-\mathbf{q}_0}{\Delta t} &= \lim_{\Delta t\to 0}\frac{1}{\Delta t}\left[\left(\cos\frac{\|\boldsymbol{\omega}\|\Delta t}{2}-1\right)+\left(\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}\sin\frac{\|\boldsymbol{\omega}\|\Delta t}{2}\right)\right] && (1.170)\\ &= \ldots && (1.171)\\ &= \lim_{\Delta t\to 0}\left[\frac{\|\boldsymbol{\omega}\|^2}{4}\frac{\Delta t}{2!}+o(\Delta t^3)+\frac{\boldsymbol{\omega}}{2}-o(\Delta t^2)\right] && (1.172)\\ &= \left[0+\frac{\boldsymbol{\omega}}{2}\right] && (1.173) \end{aligned}$$

thus yielding

$$\dot{\mathbf{q}}_W = \frac{1}{2}\begin{bmatrix} 0\\ \boldsymbol{\omega}\end{bmatrix}\circ\mathbf{q}_W(t). \tag{1.174}$$

Given that

$$\begin{aligned} \boldsymbol{\omega}_B = \mathbf{q}_{BW}\circ\boldsymbol{\omega}_W\circ\mathbf{q}_{WB} && (1.175)\\ \overline{\mathbf{q}_1\circ\mathbf{q}_2} = \overline{\mathbf{q}_2}\circ\overline{\mathbf{q}_1} && (1.176) \end{aligned}$$

it follows

$$\begin{aligned} \dot{\mathbf{q}}_{WB} &= \frac{1}{2}\boldsymbol{\omega}_W\circ\mathbf{q}_{WB} && (1.177)\\ \dot{\mathbf{q}}_{WB} &= \frac{1}{2}\mathbf{q}_{WB}\circ\boldsymbol{\omega}_B && (1.178)\\ \dot{\mathbf{q}}_{BW} &= -\frac{1}{2}\boldsymbol{\omega}_B\circ\mathbf{q}_{BW} && (1.179)\\ \dot{\mathbf{q}}_{BW} &= -\frac{1}{2}\mathbf{q}_{BW}\circ\boldsymbol{\omega}_W && (1.180) \end{aligned}$$

1.2.3 Rotational velocity and Euler angles derivatives

It is interesting and useful at this stage to understand what is the relation between the rotational velocity in the body frame $\boldsymbol{\omega}_B$ and the derivative of the roll, pitch, yaw angles (ϕ, θ, ψ).
The starting point is that

$$\dot{\mathbf{R}}_{WB} = \mathbf{R}_{WB}[\boldsymbol{\omega}_B]_\times \tag{1.181}$$

should be consistent with

$$\dot{\mathbf{R}}_{WB}(\phi, \theta, \psi) = \frac{\partial \dot{\mathbf{R}}_{WB}}{\partial \phi}\dot{\phi} + \frac{\partial \dot{\mathbf{R}}_{WB}}{\partial \theta}\dot{\theta} + \frac{\partial \dot{\mathbf{R}}_{WB}}{\partial \psi}\dot{\psi} \tag{1.182}$$

Indeed, we obtain

$$[\boldsymbol{\omega}_B]_\times = \mathbf{R}_{WB}^\top \dot{\mathbf{R}}_{WB} = \text{function of } \left(\dot{\phi}, \dot{\theta}, \dot{\psi}\right) \tag{1.183}$$

and by doing all the computations we get

$$\boldsymbol{\omega}_B = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{1.184}$$

Note that this relation is valid (with its inverse) when $\theta \neq \pm\frac{\pi}{2}$.
Interestingly, this turns equal to the following calculation:

$$\boldsymbol{\omega}_B = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x^\top(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_x^\top(\phi)\mathbf{R}_y^\top(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{1.185}$$

meaning that the derivative of the roll angle is directly mapped to the component of the rotational velocity, the derivative of the pitch angle is modified by the rotation around x, and finally the derivative of the yaw angle is modified by the rotation around x and y.
As a final consideration, we highlight that if $\theta \approx 0$ and $\phi \approx 0$ (almost hovering conditions) we obtain

$$\boldsymbol{\omega}_B = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \approx \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{1.186}$$

1.3 System actuation capability

1.3.1 Under-actuation and fully-actuation of a system

In considering UAVs we refer to a more general model where the dynamics of the agent is that of a mechanical system governed by a second order equation of the kind

$$\ddot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}, t) \tag{1.187}$$

where $\mathbf{x}$ is a generalized state vector (e.g. position and attitude) and $\mathbf{u}$ is an input/control vector (e.g. force and torque).
A system is said to be *fully actuated* if the $f(\cdot)$ map is surjective, meaning that every element of the codomain is image of at least one element of the domain, or, in other terms, that for every $\ddot{\mathbf{x}}$ there exists a $\mathbf{u}$ that produces the desired acceleration.
Relation (1.187) can be often transformed into an affine form, with respect to the input vector, thus becoming

$$\ddot{\mathbf{x}} = f_1(\mathbf{x}, \dot{\mathbf{x}}, t) + f_2(\mathbf{x}, \dot{\mathbf{x}}, t)\mathbf{u} \tag{1.188}$$

and, more interestingly, f_1 and f_2 often boil down to be matrices, allowing an easier description of the actuation space.
In this respect a useful characterization of such systems is given in the following.

Definition 9 (Fully-actuated system) *A system described by* (1.188) *is fully-actuated in configuration* $(\mathbf{x}, \dot{\mathbf{x}})$ *if it is able to command an instantaneous acceleration in an arbitrary direction of* $\mathbf{x}$*, that is, with some abuse of notation,*

$$\dim[f_2(\mathbf{x}, \dot{\mathbf{x}}, t)] = \dim(\mathbf{x}) \tag{1.189}$$

Definition 10 (Under-actuated system) *A system described by* (1.188) *is under-actuated in configuration* $(\mathbf{x}, \dot{\mathbf{x}}, t)$ *if it is not able to command an instantaneous acceleration in an arbitrary direction of* $\mathbf{x}$*, that is*

$$\dim[f_2(\mathbf{x}, \dot{\mathbf{x}}, t)] < \dim(\mathbf{x}) \tag{1.190}$$

Loosely speaking, we may say that a fully actuated system is easier to control with respect to the under-actuated one; in particular, if f_2 is full-rank, it is invertible, and allows to define the input $\mathbf{u}$ as

$$\mathbf{u} = \pi(\mathbf{x}, \dot{\mathbf{x}}, t) = f_2^{-1}(\mathbf{x}, \dot{\mathbf{x}}, t)\left[\bar{\mathbf{u}} - f_1(\mathbf{x}, \dot{\mathbf{x}}, t)\right]. \tag{1.191}$$

This leads to

$$\ddot{\mathbf{x}} = f_1(\mathbf{x}, \dot{\mathbf{x}}, t) + f_2(\mathbf{x}, \dot{\mathbf{x}}, t) f_2^{-1}(\mathbf{x}, \dot{\mathbf{x}}, t)\left[\bar{\mathbf{u}} - f_1(\mathbf{x}, \dot{\mathbf{x}}, t)\right] = \bar{\mathbf{u}} : \tag{1.192}$$

in practice, given that f_1 and f_2 are known, we observe that a fully-actuated system is feedback equivalent to $\ddot{\mathbf{x}} = \bar{\mathbf{u}}$ (while an under-actuated system is not).
Notably, the under-actuation may depend on both the state of the system and the actuation features, and the definition translates into the fact that the system cannot follow arbitrary trajectories. For example, the standard quadrotor (namely a platform with four actuators) is characterized by an under-actuated nature: it has to cope with 6 DoFs (3 translational DoFs and 3 rotational DoFs), having only 4 control inputs (see Tab. 1.2).

System	# of rotors	# of actuators	# of DoFs	Actuation
coplanar quadrotor	4	4	6	under
tilted quadrotor	4	4	6	under
tilting quadrotor	4	8	6	(potentially) full
coplanar hexarotor	6	6	6	under
tilted hexarotor	6	6	6	(potentially) full
coplanar octorotor	8	8	6	under
tilted octorotor	8	8	6	(potentially) over

Table 1.2: Actuation properties for some n-rotors of interest.

1.3.2 Constraints

A dynamical system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$ where the state $\mathbf{x}$ may refer to the the generalized position-velocity pair $(\mathbf{q}, \dot{\mathbf{q}})$ may be subject to one or more constraints described by $\Phi(\mathbf{x}, \mathbf{u}, t) \geq 0$, and in particular we consider separate constraints of the form

- $\Phi(\mathbf{u}) \geq 0$: input constraints (e.g. actuators limits), which can reduce the system to underactuation;
- $\Phi(\mathbf{q}) \geq 0$: state constraints, which can limit the dimensionality of the state space.

Moreover, we identify

- non-holonomic constraints, which can be put in the form $\Phi(\mathbf{q}, \dot{\mathbf{q}}, t) = 0$;
- holonomic constraints, which can be obtained by integration to the form $\Phi(\mathbf{q}, t) = 0$.

A non-holonomic constraint does not restrain the possible configurations that can be achieved, but the manner in which these configurations can be reached; conversely, a holonomic constraint reduces the number of degrees of freedom of the system. An example of these different constraints, is the following: a wheeled robot on a train track is subject to a holonomic constraint, while a rolling disk with no side-slip is subject to a non-holonomic constraint.

Pfaffian form

• CONSTRAINTS IN PFAFFIAN FORM

↳ linear with respect to velocity

$$\begin{cases} a_1^T(q)\,\dot q = 0 \\ a_2^T(q)\,\dot q = 0 \\ \dots \\ a_k^T(q)\,\dot q = 0 \end{cases} \qquad k < n = \dim q \quad \text{Pfaffian constraints}$$

$$\Downarrow$$

$$\boxed{A^T(q)\,\dot q = 0} \qquad A^T(q)\ (k \times n)\ \text{rank } k$$

→ k dim constraint on the velocity that must belong to a $(n-k)$ dim. space

$$\boxed{A^T(q)\,\dot q = 0} \Rightarrow \dot q \in \ker\left[A^T(q)\right]$$

$$\ker\left[A^T(q)\right] = \mathrm{span}\Big\{\underbrace{g_1(q)\cdots g_{n-k}(q)}_{n\text{-dim vectors}}\Big\}$$

$$\Rightarrow \dot q = \sum_{1}^{n-k} g_i(q)\,u_i = \underbrace{G(q)}_{n \times m}\,\underbrace{u}_{m\text{-dim}} \qquad m \triangleq n-k$$

• $u = \begin{bmatrix} u_1 \\ \dots \\ u_m \end{bmatrix}$ ⇝ linear combinators ⇝ input vector

• choice of $G(q)$ is not unique

• $u = 0 \Rightarrow \dot q = 0$ DRIFTLESS system

$$\boxed{\dot y \cos\theta = \dot x \sin\theta} \to \underbrace{\begin{bmatrix} \sin\theta & -\cos\theta & 0 \end{bmatrix}}_{A^T(q)} \underbrace{\begin{bmatrix} \dot x \\ \dot y \\ \dot\theta \end{bmatrix}}_{\dot q} = 0$$

$A^T(q) \to$ rank $= 1$

$$\rightsquigarrow \dot q \in \ker\left[A^T(q)\right] = \mathrm{span}\{g_1(q), g_2(q)\} \quad m=2$$

$$= \mathrm{span}\left\{\begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right\}$$

$$\rightsquigarrow \dot q = \sum_{1}^{2} g_i(q)\,u_i = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2$$

$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} v \\ \omega \end{bmatrix}$ ⇝ linear & angular velocities

$$\Downarrow$$

$$\dot q = G(q)\,u = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$\boxed{\begin{cases} \dot x = v\cos\theta \\ \dot y = v\sin\theta \\ \dot\theta = \omega \end{cases} \quad \text{kinematic model of the unicycle}}$$

1.4 Autonomous Guided Vehicles (AGVs)

1.4.1 The unicycle model

The unicycle is a planar vehicle with a single orientable wheel and the configuration is described in the world frame (see Fig. 1.10) by

$$\mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ \theta \end{bmatrix} \qquad \text{with} \quad \mathbf{p} \in \mathbb{R}^2 \quad \theta \in \mathbb{S}^1 \tag{1.193}$$

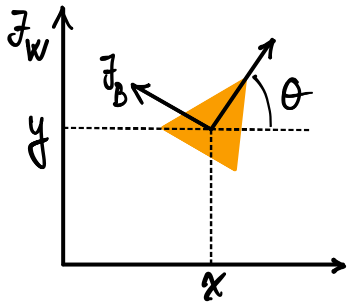


Figure 1.10: Unicycle model variables.

We can relate the translational and the rotational coordinates through the following equations

$$\dot{x} = v\cos\theta \tag{1.194}$$
$$\dot{y} = v\sin\theta \tag{1.195}$$
$$v^2 = \dot{x}^2 + \dot{y}^2 \tag{1.196}$$

which lead to

$$\begin{cases} \dot{x}\sin\theta = v\cos\theta\sin\theta \\ \dot{y}\cos\theta = v\sin\theta\cos\theta \end{cases} \quad \Rightarrow \quad \dot{x}\sin\theta - \dot{y}\cos\theta = 0. \tag{1.197}$$

This latter is a non-holonomic pure rolling constraint of the form $\Phi(\mathbf{q}, \dot{\mathbf{q}}) = 0$, which can also be written as

$$\begin{bmatrix} \sin\theta & -\cos\theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0 \qquad \Rightarrow \qquad A^\top(\mathbf{q})\dot{\mathbf{q}} = 0 \tag{1.198}$$

namely, the constraint is written in Pfaffian form: the kinematic constraint is expressed linearly with respect to the generalized velocity.
The kinematic model results from the physical interpretation of the above equation, considering the linear velocity v and the steering velocity ω.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1.199}$$

Chained form

- CHAINED FORM

the kinematic model can be put in canonical form

$\downarrow$

(2, n) chained form with 2 inputs

Def $\dot z = \gamma_1(z)\, v_1 + \gamma_2(z)\, v_2$

z ⇝ new state

$v_1\ v_2$ ⇝ new input

$$\begin{cases} \dot z_1 = v_1 \\ \dot z_2 = v_2 \\ \dot z_3 = z_2 v_1 \\ \dot z_4 = z_3 v_1 \\ \cdots \\ \dot z_n = z_{n-1} v_1 \end{cases} \Longleftrightarrow \gamma_1 = \begin{bmatrix} 1 \\ 0 \\ z_2 \\ z_3 \\ \vdots \\ z_{n-1} \end{bmatrix} \quad \gamma_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

there exist N&S conditions for transforming a 2-input driftless system in chained form

$$\Sigma_1 : \ \dot q = g_1(q)\, u_1 + g_2(q)\, u_2$$

$$\Updownarrow$$

$$\Sigma_2 : \ \dot z = \gamma_1(z)\, v_1 + \gamma_2(z)\, v_2$$

via coordinate (state) and input transform

$$z = T(q) \qquad v = \beta(q)\, u$$

If $n \leq 4$ the system can always be put in chained form

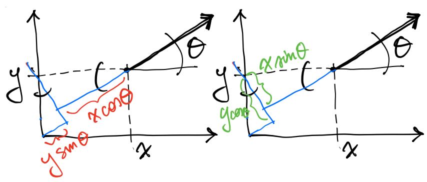
In the case of the unicycle

$$\Sigma_1 : \begin{cases} \dot x = v \cos\theta \\ \dot y = v \sin\theta \\ \dot\theta = \omega \end{cases}$$

Coordinate transformation

$$z = \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ \cos\theta & \sin\theta & 0 \\ \sin\theta & -\cos\theta & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}}_{T(q)}$$

$$\begin{cases} z_1 = \theta \\ z_2 = x\cos\theta + y \sin\theta \\ z_3 = x \sin\theta - y \cos\theta \end{cases}$$



z_1 is the orientation

z_2 is a projection on the sagittal axis

z_3 is a projection $\perp$ to the sagittal axis

⇝ $z_2\, z_3$ is a frame that is moving, with one axis (z_2) aligned with the sagittal axis

Input transformation

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & y\cos\theta - x\sin\theta \end{bmatrix}}_{\beta(q)} \underbrace{\begin{bmatrix} v \\ \omega \end{bmatrix}}_{u}$$

$$\begin{cases} v_1 = \omega \\ v_2 = v - \omega\underbrace{[x\sin\theta - y\cos\theta]}_{z_3} \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} v = v_2 + z_3 v_1 \\ \omega = v_1 \end{cases}$$

From the transformations

$$\left.\begin{cases} z_1 = \theta \\ z_2 = x\cos\theta + y\sin\theta \\ z_3 = x\sin\theta - y\cos\theta \end{cases}\right\} z = \varphi(q)$$

$$\left.\begin{cases} v = v_2 + z_3 v_1 \\ \omega = v_1 \end{cases}\right\} u = \xi(v, z)$$

$$\Downarrow$$

$$\dot z_1 = \dot\theta = \omega = v_1$$

$$\dot z_2 = (\dot x\cos\theta - x\omega\sin\theta) + (\dot y\sin\theta + y\omega\cos\theta)$$

$$= \underbrace{\dot x\cos\theta + \dot y\sin\theta}_{v} + \underbrace{\omega[y\cos\theta - x\sin\theta]}_{-\omega z_3}$$

$$= v - \omega z_3 = v_2$$

$$\dot z_3 = (\dot x\sin\theta + x\omega\cos\theta) - (\dot y\cos\theta - y\omega\sin\theta)$$

$$= \underbrace{\dot x\sin\theta - \dot y\cos\theta}_{0} + \underbrace{\omega(x\cos\theta + y\sin\theta)}_{\omega z_2}$$

$$\Downarrow$$

$$\begin{cases} \dot z_1 = v_1 \\ \dot z_2 = v_2 \\ \dot z_3 = z_2 v_1 \end{cases}$$

chained form

Differential flatness

• DIFFERENTIAL FLATNESS

Consider a non-linear dynamical system

$$\Sigma: \quad \dot{x} = f(x,u) \qquad x \in \mathbb{R}^n, \; u \in \mathbb{R}^m$$

or an input affine system

$$\Sigma: \quad \dot{x} = f(x) + G(x)\, u$$

$\Downarrow$

<u>Def</u> Σ is differentially flat if there exist a set of outputs y (flat out) such that x and u can be expressed from y and its derivatives without integration

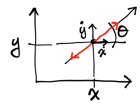
$$x = x(y, \dot{y}, \dots, y^{(r)})$$
$$u = u(y, \dot{y}, \dots, y^{(r)})$$

Given the unicycle model, how to verify the differential flatness property?

MODEL
$$\begin{cases} \dot{x} = v \cos\theta \\ \dot{y} = v \sin\theta \\ \dot{\theta} = \omega \end{cases}$$

we also have

$$\theta = \operatorname{atan2}(\dot{y}, \dot{x}) + k\pi$$



$k = 0$ forward
$k = 1$ backward

Consider the output trajectory (x_d, y_d) as a twice differentiable output

$\Downarrow$

$\boxed{\text{This output is flat}}$

$$\leadsto v_d = \pm\sqrt{\dot{x}_d^2 + \dot{y}_d^2}$$

$$\omega_d = \dot{\theta}_d = \frac{d}{dt}\operatorname{atan2}(\dot{y},\dot{x}) = \frac{\partial}{\partial \dot{x}}\operatorname{atan2}(\dot{y},\dot{x})\frac{d\dot{x}}{dt} + \frac{\partial}{\partial \dot{y}}\operatorname{atan2}(\dot{y},\dot{x})\frac{d\dot{y}}{dt} = -\frac{\dot{y}}{\dot{x}^2+\dot{y}^2}\ddot{x} + \frac{\dot{x}}{\dot{x}^2+\dot{y}^2}\ddot{y} = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\dot{x}^2+\dot{y}^2}$$

FLAT OUTPUT $(x_d \;\; y_d) \Longrightarrow$ STATE
$$\begin{cases} x_d \\ y_d \\ \theta_d = \operatorname{atan2}(\dot{y} \;\; \dot{x}) + k\pi \end{cases}$$

INPUT
$$\begin{cases} v_d = \pm\sqrt{\dot{x}_d^2 + \dot{y}_d^2} \\ \omega_d = \dfrac{\dot{x}_d \ddot{y}_d - \ddot{x}_d \dot{y}_d}{\dot{x}_d^2 + \dot{y}_d^2} \end{cases}$$

• DIFFERENTIAL FLATNESS & CHAINED FORM

General chained form for a 2-input driftless system

$$\begin{cases} \dot{z}_1 = v_1 \\ \dot{z}_2 = v_2 \\ \dot{z}_3 = z_2 v_1 \end{cases} \longrightarrow$$

state transformation
$$\begin{cases} z_1 = \theta \\ z_2 = x\cos\theta + y\sin\theta \\ z_3 = x\sin\theta - y\cos\theta \end{cases}$$

input transformation
$$\begin{cases} v = v_2 + z_3 v_1 \\ \omega = v_1 \end{cases}$$

$\Downarrow$

<u>consider z_1, z_3 as the flat outputs</u>

$$\leadsto \text{state} \quad \begin{cases} z_1 = z_1 \\ z_2 = \dfrac{\dot{z}_3}{v_1} = \dfrac{\dot{z}_3}{\dot{z}_1} \\ z_3 = z_3 \end{cases}$$

$$\leadsto \text{input} \quad \begin{cases} v_1 = \dot{z}_1 \\ v_2 = \dot{z}_2 = \dfrac{\dot{z}_1\ddot{z}_3 - \ddot{z}_1\dot{z}_3}{\dot{z}_1^2} \end{cases}$$

1.4.2 Platform control

With the unicycle, we consider three types of control:

- the tracking control;
- the position (Cartesian) and posture regulations

Tracking control

Given a feasible desired trajectory to follow

$$\mathbf{q}_d(t) = \begin{bmatrix} \mathbf{p}_d(t) \\ \theta_d(t) \end{bmatrix} = \begin{bmatrix} x_d(t) \\ y_d(t) \\ \theta_d(t) \end{bmatrix} \tag{1.200}$$

where the configuration is now a function of time and is not constant and we define the trajectory error as

$$\mathbf{e}_W = \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} \quad \Rightarrow \quad \mathbf{e}_B = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \mathbf{R}_z^\top(\theta)\mathbf{e}_W = \left[\begin{array}{cc|c} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ \hline 0 & 0 & 1 \end{array}\right] \mathbf{e}_W. \tag{1.201}$$

A general control scheme is shown in Fig. 1.11, where the flatness block is needed in order to obtain state and input from the output variables, and the feedback and feedforward actions are highlighted respectively in green (state-error feedback) and in blue (velocity reference feedforward).

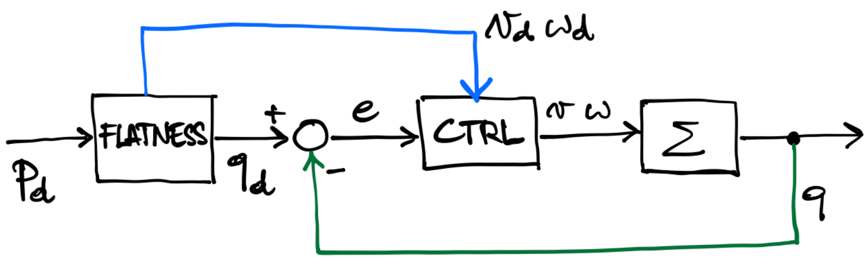


Figure 1.11: State-error trajectory tracking scheme: the feedforward loop is highlighted in blue, while the feedback loop can be seen in green.

From the equations above, it follows

$$\begin{cases} e_1 = (x_d - x)\cos\theta + (y_d - y)\sin\theta \\ e_2 = -(x_d - x)\sin\theta + (y_d - y)\cos\theta \\ e_3 = (\theta_d - \theta) \end{cases} \tag{1.202}$$

and by taking the derivative, and combining with the constraints on both the actual and the feasible trajectories

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases} \quad \text{and} \quad \begin{cases} \dot{x}_d = v_d\cos\theta_d \\ \dot{y}_d = v_d\sin\theta_d \\ \dot{\theta}_d = \omega_d \end{cases} \tag{1.203}$$

we obtain, after some calculations, a behavior for the (coupled) error dynamics

$$\left\{ \begin{array}{l} \dot{e}_1 = v_d \cos e_3 - v + e_2 \omega \\ \dot{e}_2 = v_d \sin e_3 - e_1 \omega \\ \dot{e}_3 = \omega_d - \omega \end{array} \right. . \tag{1.204}$$

With an invertible input transformation

$$\begin{array}{l} v = v_d \ \cos e_3 - u_1 \\ \omega = \omega_d - u_2 \end{array} \quad \Leftrightarrow \quad \begin{array}{l} u_1 = v_d \ \cos e_3 - v \\ u_2 = \omega_d - \omega \end{array} \tag{1.205}$$

we get

$$\dot{\mathbf{e}} = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e} + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 1 & -e_2 \\ 0 & e_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{1.206}$$

that is non-linear, coupled, and time-varying.

State-error linearization control scheme

This dynamics can be linearized around the tracking trajectory where $\mathbf{e} = 0$ into

$$\dot{\mathbf{e}} = \begin{bmatrix} 0 & \omega^{\text{des}} & 0 \\ -\omega^{\text{des}} & 0 & v^{\text{des}} \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{1.207}$$

A possible control law is then

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -k_2 & -k_3 \end{bmatrix} \mathbf{e} \tag{1.208}$$

which make the dynamics of the closed-loop error

$$\dot{\mathbf{e}} = \begin{bmatrix} -k_1 & \omega^{\text{des}} & 0 \\ -\omega^{\text{des}} & 0 & v^{\text{des}} \\ 0 & -k_2 & -k_3 \end{bmatrix} \mathbf{e}. \tag{1.209}$$

By an accurate selection of the gains k_1, k_2, k_3, we can allocate the eigenvalues of the error dynamics matrix in order to control its asymptotic convergence to zero. This can be done, for instance, by choosing

$$\left\{ \begin{array}{l} k_1 = k_3 = 2\xi a \\ k_2 = \frac{a^2 - (\omega^{\text{des}})^2}{v^{\text{des}}} \end{array} \right. \quad \text{with} \quad \xi \in (0,1) \quad \text{and} \quad a > 0 \tag{1.210}$$

where ξ and a take the roles of a damping coefficient and a natural frequency.
Note that

- because of the time-varying nature of the system (see, the velocities), this control law is not guaranteed to be asymptotically stable; nonetheless, when the linear velocity and the steering velocity are constant, this guarantee is obtained: notably, these cases correspond to when we have either circular or rectilinear trajectories;
- the control law requires $v^{\text{des}} \neq 0$: this implies that the approach is formally correct if the trajectory to be followed is persistent, and there is no stopping point or inversion of motion.

State-error non-linear control scheme

$\rightsquigarrow$ $\boxed{\text{TRAJECTORY TRACKING via State Error Feedback \#2}}$

Consider the error dynamics in the non linear form:

$$\dot{e} = \underbrace{\begin{bmatrix} \omega_d e_2 \\ -\omega_d e_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ v_d \sin e_3 \\ 0 \end{bmatrix}}_{f(e,t)} + \underbrace{\begin{bmatrix} 1 & -e_2 \\ 0 & e_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{G(e)\,u}$$

DRIFT TERM INPUT TERM

and the input transformation

$$\begin{cases} v = v_d \cos e_3 - u_1 \\ \omega = \omega_d - u_2 \end{cases}$$

② We rewrite the model as:

$$\boxed{\dot{e}_1} = \omega_d e_2 + u_1 - e_2 u_2 = \omega_d e_2 + u_1 - e_2(\omega_d - \omega) = \boxed{\omega e_2 + u_1}$$

$$\boxed{\dot{e}_2} = -\omega_d e_1 + v_d \sin e_3 + e_1 u_2 = -\omega_d e_1 + v_d \sin e_3 + e_1(\omega_d - \omega) = \boxed{v_d \sin e_3 - \omega e_1}$$

$$\boxed{\dot{e}_3 = u_2}$$

(will be used in the Lyapunov's discussion)

③ We consider a NONLINEAR version of the previous controller

$$\begin{cases} u_1 = -k_1 e_1 \\ u_2 = -k_2 e_2 - k_3 e_3 \end{cases}$$

$$\Downarrow$$

$$\begin{cases} u_1 = -\boxed{k_1(v_d, \omega_d)}\, e_1 \\ u_2 = -\boxed{\bar{k}_2}\, \underbrace{v_d \frac{\sin e_3}{e_3}}\, e_2 - \boxed{k_3(v_d, \omega_d)}\, e_3 \end{cases}$$

where

$$\left| \begin{array}{l} \left.\begin{array}{l} k_1(v_d, \omega_d) > 0 \\ k_3(v_d, \omega_d) > 0 \end{array}\right\} \text{continuous bounded with bounded deriv.} \\ \\ \bar{k}_2 > 0 \end{array} \right.$$

Choice of the control parameters $k_1, \bar{k}_2, k_3$:

$\rightsquigarrow$ from the approximate linearization case:

$$\boxed{\begin{array}{l} k_1 = k_3 = 2\zeta a \\ k_2 = \dfrac{a^2 - \omega_d^2}{v_d} \end{array}} \rightsquigarrow \sqrt{v_d k_2 + \omega_d^2} = a$$

$$\Downarrow$$

$$\bar{k}_2 = b$$

$$k_1 = k_3 = 2\zeta \sqrt{b\, v_d(t) + \omega_d^2(t)}$$

with $b > 0$

$\zeta \in (0,1)$

TH Assuming that v_d and ω_d are bounded with bounded derivatives and that $v_d(t) \not\to 0$ or $\omega_d \not\to 0$ for $t \to \infty$ (they do not both converge to zero)

then

the control law above GLOBALLY ASYMPTOTICALLY stabilizes $e = 0$

Proof is based on Lyapunov arguments

Proof

Choose Lyapunov function

$$\boxed{V = \frac{\bar{k}_2}{2}\left(e_1^2 + e_2^2\right) + \frac{1}{2} e_3^2}$$

$V > 0$ in a neighborhood of the origin
$V = 0$ at $e = 0$
V radially unbounded

$$\begin{aligned}
\dot{V} &= 2\,\frac{\bar{k}_2}{2}\left(e_1\dot{e}_1 + e_2\dot{e}_2\right) + e_3\dot{e}_3 \\
&= \bar{k}_2 e_1\left(e_2\omega - k_1 e_1\right) + \\
&\quad + \bar{k}_2 e_2\left(v_d \sin e_3 - e_1\omega\right) + \\
&\quad + e_3\left(-\bar{k}_2 v_d \frac{\sin e_3}{e_3} e_2 - k_3 e_3\right) \\
&= \cancel{\bar{k}_2 e_1 e_2 \omega} - \bar{k}_2 k_1 e_1^2 + \\
&\quad + \cancel{\bar{k}_2 v_d e_2 \sin e_3} - \cancel{\bar{k}_2 \omega e_2 e_1} + \\
&\quad - \cancel{\bar{k}_2 v_d e_2 \sin e_3} - k_3 e_3^2
\end{aligned}$$

$$\boxed{\dot{V} = -k_1\bar{k}_2 e_1^2 - k_3 e_3^2 \leq 0}$$

⇝ we cannot use LaSalle theorem (quantities are also time varying)

⇝ we can resort to Barbalat Lemma since

V is lower bounded
$\dot{V} \leq 0$
$\ddot{V}$ is bounded (e is bounded)

$$\Downarrow$$

$$\boxed{\dot{V} \longrightarrow 0} \implies \begin{array}{c} e_1, e_3 \longrightarrow 0 \\ + \\ (v_d^2 + \omega_d^2)\, e_2^2 \longrightarrow 0 \end{array} \qquad \square$$

__Note:__ it is required a persistent state trajectory therefore ω_d and v_d __cannot__ be both zero ⇝ v_d can converge to zero provided that ω_d does not (degeneracy of the trajectory to a rotation on the spot)

The actual realization of these control schemes is shown in Fig. 1.12.

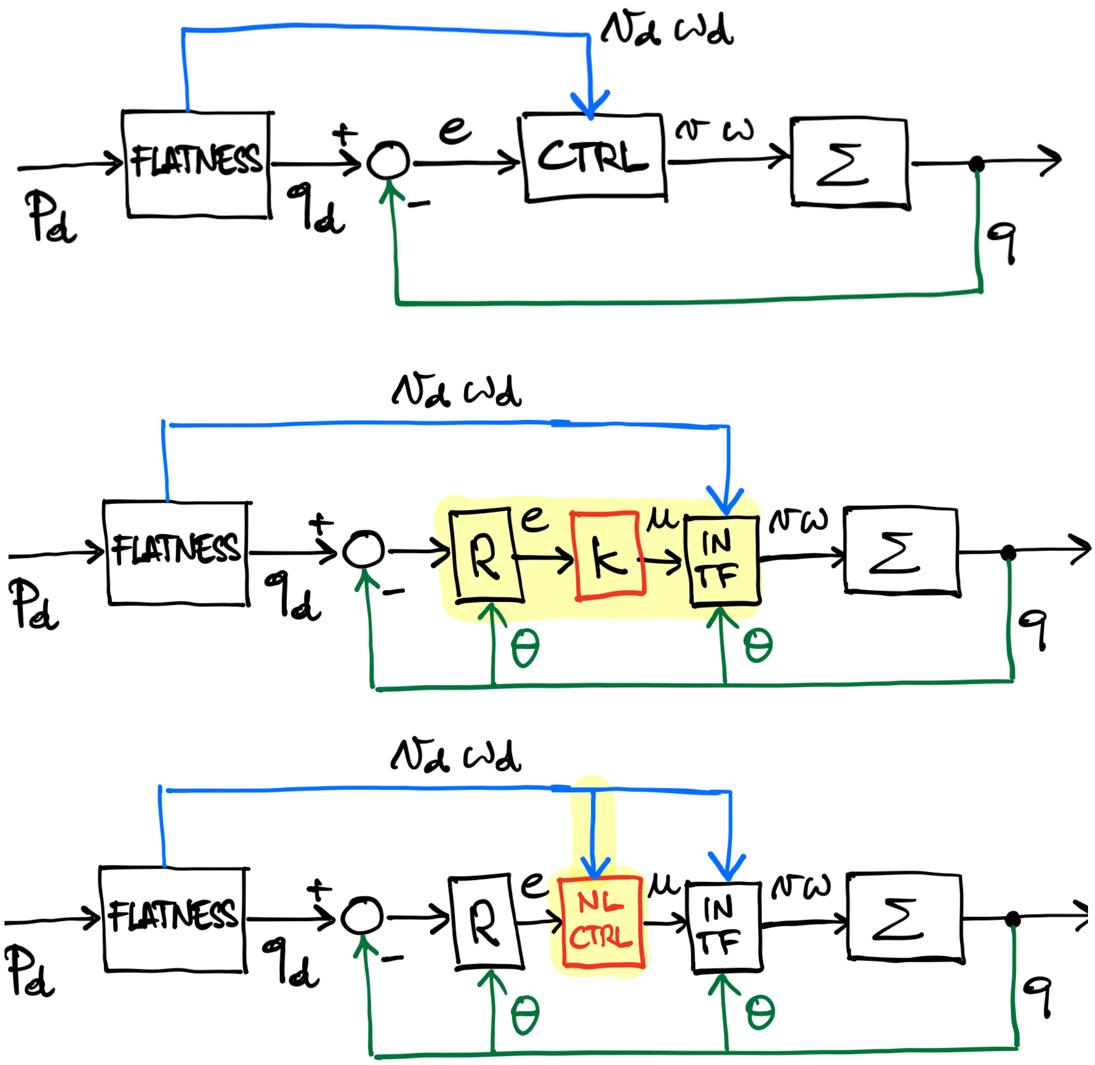


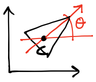
Figure 1.12: From top to bottom: the general state-error feedback control scheme; control scheme with linearization of the error dynamics; tracking scheme with non-linear controller.

Output-error feedback control schemes

<u>Rationale</u>

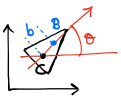
- find an INVERTIBLE map between the input and some derivative of the output
- make the system linear by inverting this map

$\rightsquigarrow$ map between the velocity input and the Cartesian output



$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \underbrace{\begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\end{bmatrix}}_{|\cdot| = 0 \rightsquigarrow \text{singular!}} \begin{bmatrix}v\\ \omega\end{bmatrix}$$

$$\Downarrow$$



$$\begin{cases} y_1 = x + b\cos\theta \\ y_2 = y + b\sin\theta \end{cases}$$

(simplest) change of variables

$$\begin{cases} \dot{y}_1 = \dot{x} - b\sin\theta\,\dot{\theta} = v\cos\theta - \omega b\sin\theta \\ \dot{y}_2 = \dot{y} + b\cos\theta\,\dot{\theta} = v\sin\theta + \omega b\cos\theta \end{cases}$$

$$\begin{bmatrix}\dot{y}_1\\ \dot{y}_2\end{bmatrix} = \underbrace{\begin{bmatrix}\cos\theta & -b\sin\theta\\ \sin\theta & b\cos\theta\end{bmatrix}}_{|\cdot| = b \rightsquigarrow \underline{\text{non}}\text{ singular if } b \neq 0} \begin{bmatrix}v\\ \omega\end{bmatrix}$$

$\rightsquigarrow$ $\begin{bmatrix}\dot{y}_1\\ \dot{y}_2\end{bmatrix} = T(\theta)\begin{bmatrix}v\\ \omega\end{bmatrix}$

this suggest the following input tf

$$\begin{bmatrix}u_1\\ u_2\end{bmatrix} = T(\theta)\begin{bmatrix}v\\ \omega\end{bmatrix} \Rightarrow T(\theta) = \begin{bmatrix}\cos\theta & -b\sin\theta\\ \sin\theta & b\cos\theta\end{bmatrix}$$

$$\rightsquigarrow$$

$$\begin{bmatrix}v\\ \omega\end{bmatrix} = T^{-1}(\theta)\begin{bmatrix}u_1\\ u_2\end{bmatrix} \Rightarrow T^{-1}(\theta) = \begin{bmatrix}\cos\theta & \sin\theta\\ -\frac{\sin\theta}{b} & \frac{\cos\theta}{b}\end{bmatrix}$$

$\rightsquigarrow$ with this change of input the system reads

$$\begin{cases} \boxed{\begin{aligned}\dot{y}_1 &= u_1\\ \dot{y}_2 &= u_2\end{aligned}} \\ \dot{\theta} = -\frac{\sin\theta}{b}u_1 + \frac{\cos\theta}{b}u_2 \end{cases}$$

ω

- the $\boxed{\text{system}}$ shows decoupled dynamics in the two components, which are referring to the Cartesian position
- the orientation evolves according to the third equation

$\rightsquigarrow$ linear controller (PD)

$$\begin{cases} u_1 = k_1(y_{1d} - y_1) + \dot{y}_{1d} \\ u_2 = k_2(y_{2d} - y_2) + \dot{y}_{2d} \end{cases}$$

with $k_1, k_2 > 0$

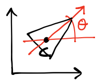
<u>Note 1</u>: θ is not controlled because with this scheme we are doing output error feedback (on the Cartesian position) θ evolves according to eq. before...

<u>Note 2</u>: the reference trajectory must be given in terms of $y_1\ y_2$, that is the position of B: this may be arbitrary and does not have smoothness requirements (square angles!)... as far as $\underline{b \neq 0}$

Rationale

- find an INVERTIBLE map between the input and some derivative of the output
- make the system linear by inverting this map

$\rightsquigarrow$ map between the velocity input and the Cartesian output



$$\begin{bmatrix}\dot x\\ \dot y\end{bmatrix} = \underbrace{\begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\end{bmatrix}}_{|\cdot| = 0 \ \rightsquigarrow\ \text{singular!}} \begin{bmatrix} v\\ \omega\end{bmatrix}$$

$$\Downarrow$$

only v affects $\dot p$ and we cannot recover ω from first-order differential information

$\rightsquigarrow$ Consider the linear acceleration is available : $\boxed{a = \dot v}$

$$\begin{bmatrix}\ddot x\\ \ddot y\end{bmatrix} = \begin{bmatrix}\dot v \cos\theta - v \sin\theta\, \dot\theta\\ \dot v \sin\theta + v \cos\theta\, \dot\theta\end{bmatrix}$$

$$\begin{bmatrix}\ddot x\\ \ddot y\end{bmatrix} = \underbrace{\begin{bmatrix}\cos\theta & -v\sin\theta\\ \sin\theta & v\cos\theta\end{bmatrix}}_{|\cdot| = v\cos^2\theta + v\sin^2\theta = v \text{, non singular if } v\neq 0}\begin{bmatrix}\dot v\\ \dot\theta\end{bmatrix}$$

$$\Downarrow$$

- Choose an output to which a desired behaviour can be assigned $\rightarrow (x \ y)$
- Differentiate the output until the input appears in a non singular way $\rightarrow (\dot v \ \dot\theta)$ $\downarrow$ (a, ω)

Input transformation

$$\begin{bmatrix}\ddot x\\ \ddot y\end{bmatrix} = T(v,\theta)\begin{bmatrix}a\\ \omega\end{bmatrix}$$

this suggest the following input tf

$$\begin{bmatrix}u_1\\ u_2\end{bmatrix} = T(v,\theta)\begin{bmatrix}a\\ \omega\end{bmatrix} \Rightarrow T = \begin{bmatrix}c\theta & -v\, s\theta\\ s\theta & v\, c\theta\end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix}a\\ \omega\end{bmatrix} = T^{-1}(v,\theta)\begin{bmatrix}u_1\\ u_2\end{bmatrix} \Rightarrow T^{-1} = \begin{bmatrix}c\theta & s\theta\\ -\frac{s\theta}{v} & \frac{c\theta}{v}\end{bmatrix}$$

(u_1, u_2) are the new input related to $(a \ \omega)$ through $T^{-1}(v,\theta)$

$\downarrow$

they are the second derivative of the output trajectory ...

The system becomes

$$\begin{cases} z_1 = x\\ z_2 = y\\ z_3 = \dot x = v\cos\theta\\ z_3 = \dot y = v\sin\theta\end{cases}$$

$$\Downarrow$$

$$\begin{cases} \dot z_1 = \dot x = v\cos\theta\\ \dot z_2 = \dot y = v\sin\theta\\ \boxed{\begin{aligned}\dot z_3 &= \ddot x = u_1\\ \dot z_4 &= \ddot y = u_2\end{aligned}} \longrightarrow \text{decoupled dynamics}\end{cases}$$

$$\Downarrow$$

$$\boxed{\begin{aligned}\ddot z_1 &= u_1\\ \ddot z_2 &= u_2\end{aligned}}$$ DOUBLE INTEGRATOR MODEL

$\rightsquigarrow$ linear controller (PD)

$$u_1 = k_{p1}(x_d - x) + k_{d1}(\dot{x}_d - \dot{x}) + \ddot{x}_d$$

$$u_2 = k_{p2}(y_d - y) + k_{d2}(\dot{y}_d - \dot{y}) + \ddot{y}_d$$

with $k_{p*}\ k_{d*} > 0$

Note 1 since $z_1 = x \quad z_3 = \dot{x}$
$z_2 = y \quad z_4 = \dot{y}$

$$u_1 \rightsquigarrow \left.\begin{array}{l}(z_{1d} - z_1)\\ (z_{3d} - z_3)\\ \dot{z}_{3d}\end{array}\right\} PD$$

$$u_2 \rightsquigarrow \left.\begin{array}{l}(z_{2d} - z_2)\\ (z_{4d} - z_4)\\ \dot{z}_{4d}\end{array}\right\} PD$$

Note 2 the desired trajectory must be smooth and persistent

smooth $\Rightarrow$ $x_d\ \dot{x}_d\ \ddot{x}_d$, $y_d\ \dot{y}_d\ \ddot{y}_d$

persistent $\Rightarrow$ $v_d \neq 0$

The actual realization of these control schemes is shown in Fig. 1.13.

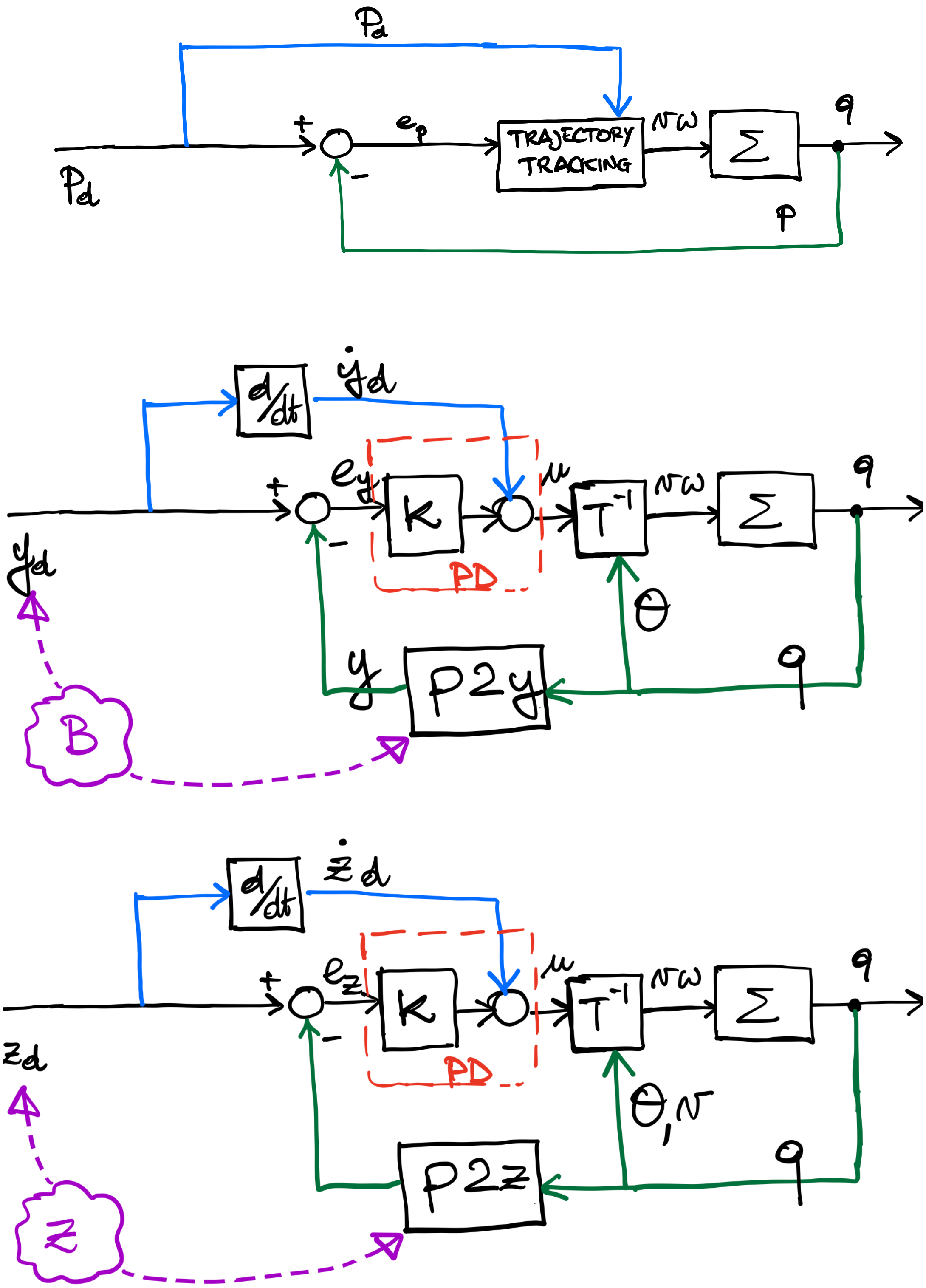


Figure 1.13: From top to bottom: the general output-error feedback control scheme; control scheme with feedback linearization based on a reference point on the sagittal axis; control scheme with feedback linearization based on second order derivatives.

Cartesian and posture control

In this case the task is to drive the unicycle to a desired configuration in terms of

- position $\mathbf{p}$ regardless the orientation θ (Cartesian regulation/control);
- full position and orientation configuration (posture regulation/control).

Cartesian control

Without loss of generality, we consider the following desired target configuration

$$\mathbf{q}^{\text{des}} = \begin{bmatrix} \mathbf{p}^{\text{des}} \\ \forall \end{bmatrix} = \begin{bmatrix} x^{\text{des}} \\ y^{\text{des}} \\ \forall \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \forall \end{bmatrix} \tag{1.211}$$

and, with respect to the current position (x, y) and orientation θ, we define the position error vector $\mathbf{e_p}$ and the unit vector sagittal axis $\mathbf{n}$

$$\mathbf{e_p} = \begin{bmatrix} 0 - x \\ 0 - y \end{bmatrix} = \begin{bmatrix} -x \\ -y \end{bmatrix} \qquad \mathbf{n} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}. \tag{1.212}$$

The projection of the error on the sagittal axis is (see Fig. 1.4.2)

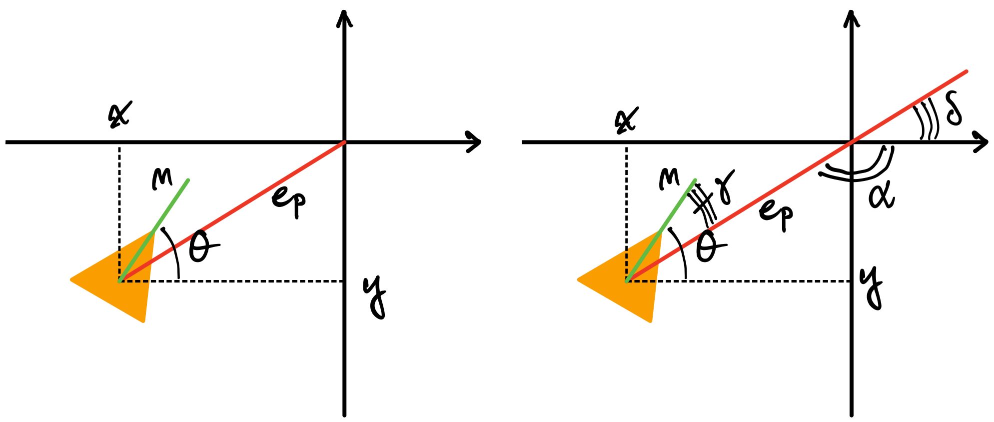


Figure 1.14: Posture regulation: vector and angle definition.

$$< \mathbf{e_p}, \mathbf{n} >= \mathbf{e_p}^\top \mathbf{n} = -x\cos\theta - y\sin\theta \tag{1.213}$$

and the aim of the position regulation is to take this projection to zero.
Observing the angles (see Fig. 1.4.2), we have that $\alpha = \text{atan2}(y, x)$ and $\delta = \alpha + \pi$, and finally the angle between the position error vector and the sagittal axis is

$$\gamma = \delta - \theta = \text{atan2}(y, x) + \pi - \theta : \tag{1.214}$$

the aim of the controller is to take this angle to zero.
The control law that we adopt acts on the velocity pair (v, ω) and is the following

$$v = k_v \mathbf{e_p}^\top \mathbf{n} = k_v \left(-x \ \cos\theta - y \ \sin\theta\right) \tag{1.215}$$

$$\omega = k_\omega \gamma = k_\omega \left(\text{atan2}(y, x) + \pi - \theta\right) \tag{1.216}$$

where k_v and k_ω are positive constant gains.

Proof via Lyapunov arguments

Consider the Lyapunov function

- $V = \frac{1}{2}(x^2 + y^2) \geq 0$
- $\dot{V} = \frac{1}{2}(2x\dot{x} + 2y\dot{y})$

$$= x\dot{x} + y\dot{y}$$

$$= x\,v\cos\theta + y\,v\sin\theta$$

$$= -k_v(x\cos\theta + y\sin\theta)^2 \leq 0$$

$\rightsquigarrow$ $x\cos\theta + y\sin\theta = 0$ not only when $(x, y) = \bar{0}$
therefore we cannot resort to La Salle ...

$\rightsquigarrow$ we make use of Barbalat's Lemma
observing that $\ddot{V}$ is bounded:

$$\lim_{t \to \infty} \dot{V} = 0 \implies x\cos\theta + y\sin\theta \to 0$$

\+
feedback on ω allows
to align ep to sagittal axis

$\square$

Posture control

Without loss of generality, we consider the following desired target configuration

$$\mathbf{q}^{\text{des}} = \begin{bmatrix} \mathbf{p}^{\text{des}} \\ \theta^{\text{des}} \end{bmatrix} = \begin{bmatrix} x^{\text{des}} \\ y^{\text{des}} \\ \theta^{\text{des}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.217}$$

and, with respect to the current position (x, y) and orientation θ, we have the position error vector $\mathbf{e_p}$ and the unit vector sagittal axis $\mathbf{n}$

$$\mathbf{e_p} = \begin{bmatrix} 0 - x \\ 0 - y \end{bmatrix} = \begin{bmatrix} -x \\ -y \end{bmatrix} \qquad \mathbf{n} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}. \tag{1.218}$$

Also, the length of the error vector $\mathbf{e_p}$ is

$$\rho = \sqrt{x^2 + y^2} \tag{1.219}$$

which is combined with the angles γ and δ

$$\gamma = \text{atan2}(y, x) + \pi - \theta \tag{1.220}$$

$$\delta = \gamma + \theta \tag{1.221}$$

to provide a polar representation of the unicycle pose: the aim of the regulation action is to bring these quantities to zero. The derivative results, after some computation, as

$$\dot{\rho} = -v\cos\gamma \tag{1.222}$$

$$\dot{\gamma} = \frac{\sin\gamma}{\rho} v - \omega \tag{1.223}$$

$$\dot{\delta} = \frac{\sin\gamma}{\rho} v \tag{1.224}$$

highlighting a singularity for $\rho = 0$, namely at the origin.
A possible control law is

$$v = k_v \ \rho \ \cos\gamma \tag{1.225}$$

$$\omega = k_\omega \ \gamma + k_v \frac{\sin\gamma \ \cos\gamma}{\gamma} (\gamma + k_\delta \ \delta) \tag{1.226}$$

where the former control modulates v and is related to the projection of $\mathbf{e_p}$, similarly to the action taken in the position regulation, and the latter shows a new term to take into the account the orientation error.

Proof via Lyapunov arguments

$$V = \frac{1}{2}\left(\rho^2 + \gamma^2 + k_3 \delta^2\right) > 0$$

$\rho \neq 0 \longrightarrow$ singularity at the origin

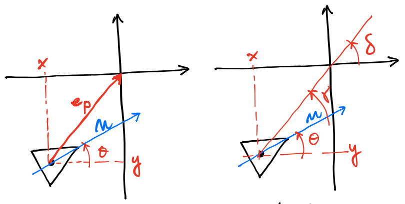
$$\begin{aligned}
\dot V &= \rho\dot\rho + \gamma\dot\gamma + k_3\,\delta\dot\delta \\
&= -\rho v\cos\gamma + \gamma\frac{\sin\gamma}{\rho}v - \gamma\omega + k_3\,\delta\frac{\sin\gamma}{\rho}v \\
&= -k_1\rho^2\cos^2\gamma + k_1\cancel{\gamma\sin\gamma\cos\gamma} + \cancel{k_1k_3\,\delta\sin\gamma\cos\gamma} \\
&\quad - \gamma^2 k_2 - k_1\sin\gamma\cos\gamma(\cancel{\gamma} + \cancel{k_3\delta}) \\
&= -k_1\rho^2\cos^2\gamma - \gamma^2 k_2 \leq 0
\end{aligned}$$

Barbalat's Lemma : $\ddot V$ is bounded

$$\lim_{t\to\infty} \dot V = 0$$

it can also be shown that ρ, γ, δ converge to zero

$\square$



the change of variable is always well defined apart from the origin...

$$x = y = 0 \Rightarrow \delta \text{ is not defined}$$

Introduce the input variable

$$\boxed{u = v/\rho}$$ (not defined at the origin)

the former dynamics

$$\begin{cases} \dot\rho = -v\cos\gamma \\ \dot\gamma = \frac{\sin\gamma}{\rho} v - \omega \\ \dot\delta = \frac{\sin\gamma}{\rho} v \end{cases}$$

is now modified as:

$$\begin{cases} \dot\rho = -\rho u \cos\gamma \\ \dot\gamma = \sin\gamma \cdot u - \omega \\ \dot\delta = \sin\gamma \cdot u \end{cases}$$

and the control law (as before)

$$u = k_1 \cos\gamma$$

$$\omega = k_2\gamma + k_1 \frac{\sin\gamma\cos\gamma}{\gamma}(\gamma + k_3\delta)$$

<u>Proof via Lyapunov arguments</u>

$$V = \frac{1}{2}(\rho^2 + \gamma^2 + k_3\delta^2) \quad \text{(as before)}$$

$$\begin{aligned} \dot V &= \rho\dot\rho + \gamma\dot\gamma + k_3\delta\dot\delta \\ &= -\rho^2 u\cos\gamma + \gamma \cdot u\sin\gamma - \gamma\omega + k_3\delta u \sin\gamma \\ &= -\rho^2 k_1\cos^2\gamma + \cancel{\gamma k_1 \cos\gamma\sin\gamma} + \\ &\quad + \cancel{k_3\delta k_1 \cos\gamma\sin\gamma} + \\ &\quad - k_2\gamma^2 - k_1 \sin\gamma\cos\gamma(\cancel{\gamma} + \cancel{k_3\delta}) \\ &= -\rho^2 k_1 \cos^2\gamma - k_2\gamma^2 \leq 0 \end{aligned}$$

In the set $R = \{\bar q \text{ s.t. } \dot V(\bar q) = 0\}$

we have $\bar q$: $\underbrace{\bar\rho = 0, \ \bar\gamma = 0}$

$$\Downarrow$$

$$\bar u = k_1 \cos\gamma|_{\bar\gamma=0} = k_1$$

$$\bar\omega = \ldots|_{\bar\gamma=0} = k_1 k_3 \bar\delta$$

$$\begin{cases} \dot{\bar\rho} = -\bar\rho\bar u \cos\bar\gamma = 0 \\ \dot{\bar\gamma} = \sin\bar\gamma \cdot \bar u - \bar\omega = k_1 k_3 \bar\delta \\ \dot{\bar\delta} = \sin\bar\gamma \cdot \bar u = 0 \end{cases}$$

$\Rightarrow$ also $\bar\delta = 0$

<u>La Salle</u> $\rightsquigarrow$ the origin is a asymptotically stable equilibrium point □

The actual realization of the regulation schemes is shown in Fig. 1.15.

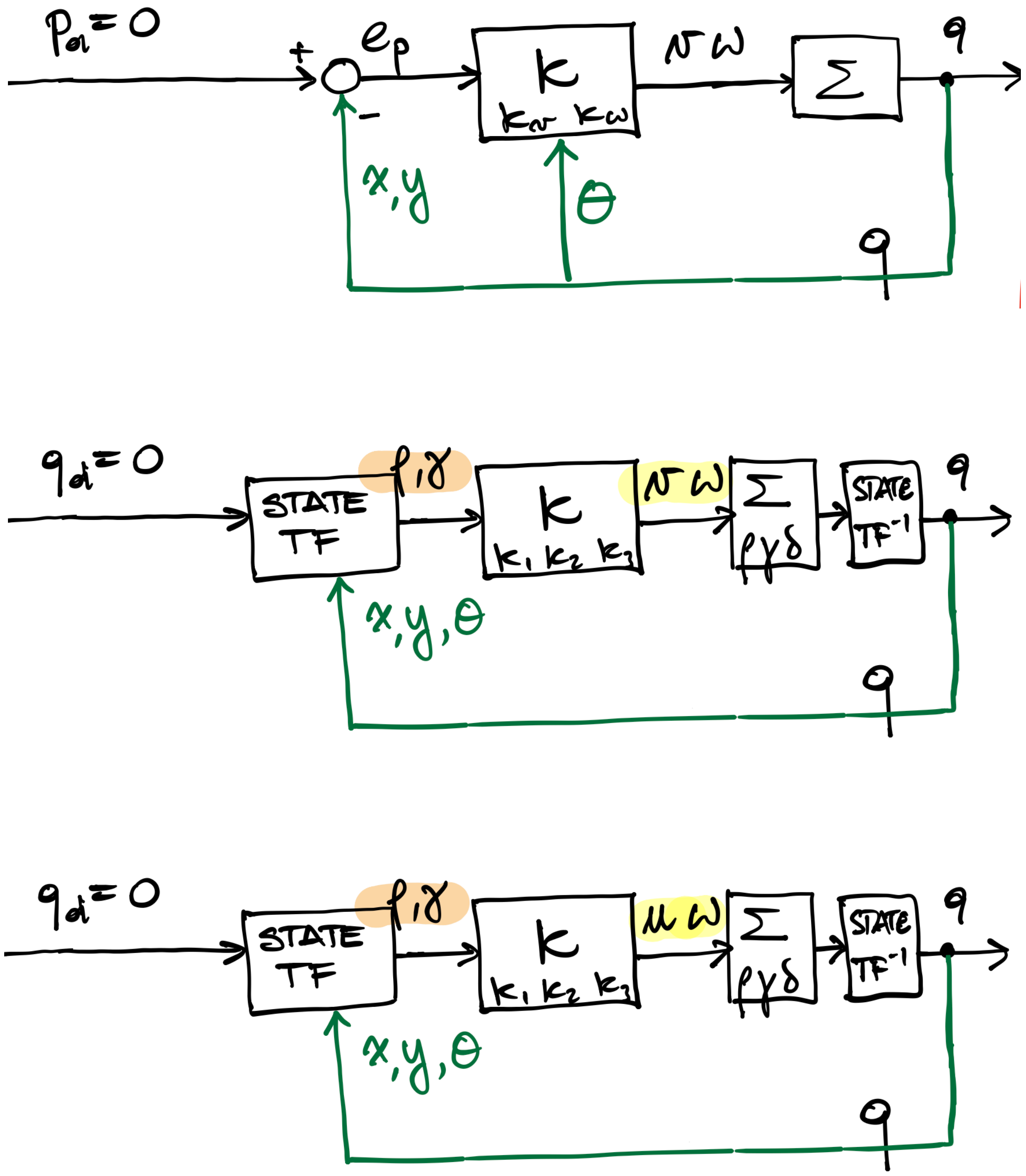


Figure 1.15: From top to bottom: Cartesian regulation scheme; posture regulation scheme (with singularity at the origin); posture regulation scheme (without singularity at the origin).

1.5 Unmanned Aerial Systems (UAVs)

1.5.1 The standard n-rotor model

The standard n-rotor is made of a rigid body with n propellers spinning about their own axis (including the special cases of all parallel or all different axes). In particular, the number n of propellers and the axes mutual orientations determine if the n-rotor is an under-actuated or fully-actuated system.
To describe the dynamics of the n-rotor, we introduce the *body frame* $\mathcal{F}_B$, whose origin O_B coincides with the CoM of the platform, and the inertial *world frame* $\mathcal{F}_W$: the position of O_B in $\mathcal{F}_W$ and the orientation of $\mathcal{F}_B$ with respect to $\mathcal{F}_W$ are respectively denoted by the vector $\mathbf{p} \in \mathbb{R}^3$ and by the rotation matrix $\mathbf{R} \in \mathbb{SO}(3)$, hence the pair $\mathcal{X} = (\mathbf{p}, \mathbf{R}) \in \mathbb{R}^3 \times \mathbb{SO}(3)$ describes the full-pose of the vehicle in $\mathcal{F}_W$.
The twist of the platform is indicated by the pair $(\mathbf{v}, \boldsymbol{\omega})$ where $\mathbf{v} = \dot{\mathbf{p}} \in \mathbb{R}^3$ denotes the linear velocity of O_B in $\mathcal{F}_W$, and $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of $\mathcal{F}_B$ with respect to $\mathcal{F}_W$, expressed in $\mathcal{F}_B$. Thus, the kinematics is governed by the relation

$$\dot{\mathbf{p}} = \mathbf{v} \tag{1.227}$$

$$\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}]_\times \tag{1.228}$$

where for the last expression we recall the expression of the tangent plane to $\mathbb{SO}(3)$ at $\mathbf{R}$.
The motion equations are derived using the standard Newton-Euler approach for the dynamics and considering the forces and torques that are generated by each propeller. The i-th propeller, with $i = 1 \dots n$, rotates around its own spinning axis passing through the center O_{P_i} with a controllable spinning rate $\omega_i \in \mathbb{R}$. There exist two kinds of propellers, spinning clockwise (CW) or counterclockwise (CCW) with respect to their axis ($\hat{\mathbf{u}}_{z_i}$). If the propeller is CW then its angular velocity in $\mathcal{F}_B$ is $-\omega_i \hat{\mathbf{u}}_{z_i}$, otherwise is $+\omega_i \hat{\mathbf{u}}_{z_i}$. We define $u_i = \omega_i |\omega_i| \in \mathbb{R}$ as the control input. According to the most commonly accepted model, the propeller applies at O_{P_i}

1. a thrust force $\mathbf{f}_i \in \mathbb{R}^3$ that, expressed in $\mathcal{F}_B$, is equal to

$$\mathbf{f}_i = c_{f_i} u_i \hat{\mathbf{u}}_{z_i} \tag{1.229}$$

   where $c_{f_i} \in \mathbb{R}^+$ is a constant parameter.

2. a drag moment $\boldsymbol{\tau}_i^d \in \mathbb{R}^3$ whose direction is opposite to the angular velocity of the propeller and whose expression in $\mathcal{F}_B$ is

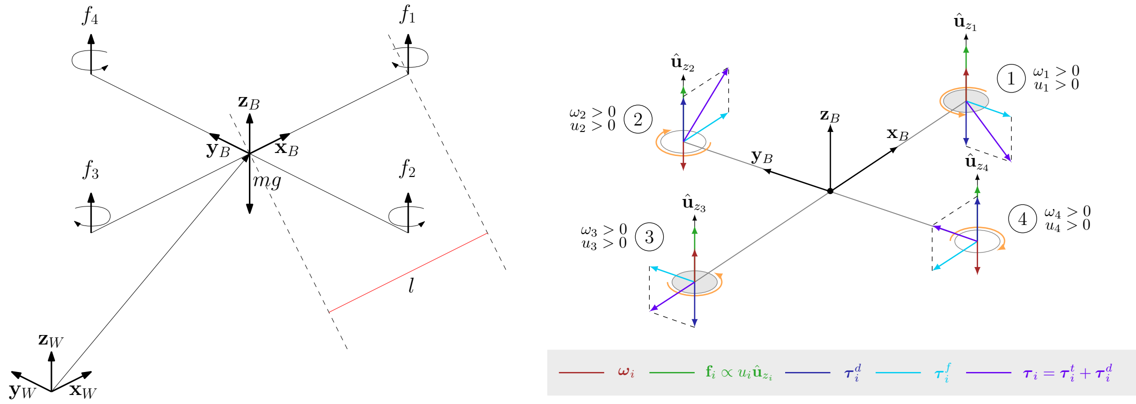
$$\boldsymbol{\tau}_i^d = c_{\tau_i} u_i \hat{\mathbf{u}}_{z_i} \tag{1.230}$$

   where $c_{\tau_i} \in \mathbb{R}$ is a constant parameter (positive if the propeller is CW and negative otherwise).

Considering now the entire UAV (see Fig. 1.16(a), for a quadcopter example), $\boldsymbol{\tau}_i^t = \mathbf{p}_i \times \mathbf{f}_i \in \mathbb{R}^3$ represents the thrust moment associated to the i-th propeller, and the total control force $\mathbf{f}_c \in \mathbb{R}^3$ and the total control moment $\boldsymbol{\tau}_c \in \mathbb{R}^3$ applied at O_B and expressed in $\mathcal{F}_B$ are

$$\mathbf{f}_c = \textstyle\sum_{i=1}^n \mathbf{f}_i = \sum_{i=1}^n c_{f_i} \hat{\mathbf{u}}_{z_i} u_i \tag{1.231}$$

$$\boldsymbol{\tau}_c = \textstyle\sum_{i=1}^n (\boldsymbol{\tau}_i^t + \boldsymbol{\tau}_i^d) = \sum_{i=1}^n \left(c_{f_i} \mathbf{p}_i \times \hat{\mathbf{u}}_{z_i} + c_{\tau_i} \hat{\mathbf{u}}_{z_i}\right) u_i \tag{1.232}$$



(a) Reference frames

(b) Forces and torques

Figure 1.16: Schematic picture of an UAV.

A summary of forces and torques acting on a quadcopter is given as an example in Fig. 1.16(b). Introducing the control input vector $\mathbf{u} = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}^\top \in \mathbb{R}^n$, (1.231) and (1.232) can be shortened as

$$\mathbf{f}_c = \mathbf{F}\mathbf{u}, \quad \text{and} \quad \boldsymbol{\tau}_c = \mathbf{M}\mathbf{u}, \tag{1.233}$$

where the *control force input matrix* $\mathbf{F} \in \mathbb{R}^{3\times n}$ and the *control moment input matrix* $\mathbf{M} \in \mathbb{R}^{3\times n}$ depend on the geometric and aerodynamic parameters introduced before.
Neglecting the second order effects (such as, the gyroscopic and inertial effects due to the rotors and the flapping) the dynamics of the n-rotor is described by the following system of Newton-Euler equations

$$m\ddot{\mathbf{p}} = -mg\mathbf{e}_3 + \mathbf{R}\mathbf{f}_c = -mg\mathbf{e}_3 + \mathbf{R}\mathbf{F}\mathbf{u} \tag{1.234}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau}_c = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}\mathbf{u}, \tag{1.235}$$

where $g > 0$, $m > 0$ and $\mathbf{J} \in \mathbb{R}^{3\times 3}$ are the gravitational acceleration, the total mass of the platform and its positive definite inertia matrix, respectively, and $\mathbf{e}_i$ is the i-th canonical basis vector of $\mathbb{R}^3$ with $i = 1, 2, 3$.

Example 7 *If we consider the general case of the quadrotor with tilted propellers (propellers are tilted with an angle α along the axis orthogonal to the arm of length l) we get for the $\mathbf{F}$ and $\mathbf{M}$ matrices the following expressions:*

$$\mathbf{F} = c_f \cdot \begin{bmatrix} 0 & s\alpha & 0 & -s\alpha \\ s\alpha & 0 & -s\alpha & 0 \\ c\alpha & c\alpha & c\alpha & c\alpha \end{bmatrix} \tag{1.236}$$

$$\mathbf{M} = l \cdot c_f \cdot \begin{bmatrix} 0 & c\alpha & 0 & -c\alpha \\ -c\alpha & 0 & c\alpha & 0 \\ s\alpha & -s\alpha & s\alpha & -s\alpha \end{bmatrix} + |c_\tau| \cdot \begin{bmatrix} 0 & s\alpha & 0 & -s\alpha \\ -s\alpha & 0 & s\alpha & 0 \\ -c\alpha & c\alpha & -c\alpha & c\alpha \end{bmatrix} \tag{1.237}$$

where the usual notation $c\alpha = \cos(\alpha)$ and $s\alpha = \sin(\alpha)$ has been adopted.
In the case of the standard coplanar quadrotor, $\alpha = 0$ and the computation above gives:

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ c_f & c_f & c_f & c_f \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0 & l \cdot c_f & 0 & -l \cdot c_f \\ -l \cdot c_f & 0 & l \cdot c_f & 0 \\ -|c_\tau| & |c_\tau| & -|c_\tau| & |c_\tau| \end{bmatrix} \tag{1.238}$$

1.5.2 System coupling and actuation properties

It is useful at this stage to provide a unified overview of the kinematics and dynamics equations both with a $\mathcal{X} = (\mathbf{p}, \mathbf{R}) \in \mathbb{R}^3 \times \mathbb{SO}(3)$ representation (as given in detail above):

$$\dot{\mathbf{p}} = \mathbf{v} \tag{1.239}$$
$$\dot{\mathbf{R}} = \mathbf{R}[\omega]_\times \tag{1.240}$$
$$m\ddot{\mathbf{p}} = -mg\mathbf{e}_3 + \mathbf{RFu} \tag{1.241}$$
$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{Mu} \tag{1.242}$$

and with a $\mathcal{X} = (\mathbf{p}, \mathbf{q}) \in \mathbb{R}^3 \times \mathbb{S}^3$, where we use the quaternion representation for the pose orientation and which can be retrieved from the previous analysis by replacing the rotation matrices and employing the quaternion algebra:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{1.243}$$
$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2}M(\mathbf{q}) \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \tag{1.244}$$
$$m\ddot{\mathbf{p}} = -mg\mathbf{e}_3 + \mathbf{R}(\mathbf{q})\mathbf{Fu} \tag{1.245}$$
$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{Mu} \tag{1.246}$$

We can here highlight how the rotational dynamics (1.242)-(1.246) influences through (1.240)-(1.244) the translational dynamics (1.241)-(1.245). This cascaded dependency of the translational dynamics on the rotation one implies the emergence of a coupling between the control force and the control torque.
If we go back to the introduction on partially and fully actuated systems (Sec. 1.3.1), we can see that the affine form (1.188) corresponds to

$$\begin{bmatrix} \ddot{\mathbf{p}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} -g\mathbf{e}_3 \\ -\mathbf{J}^{-1}(\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}) \end{bmatrix} + \begin{bmatrix} m^{-1}\mathbf{RF} \\ \mathbf{J}^{-1}\mathbf{M} \end{bmatrix} \mathbf{u} \tag{1.247}$$

Full actuation is attained if $\begin{bmatrix} m^{-1}\mathbf{RF} \\ \mathbf{J}^{-1}\mathbf{M} \end{bmatrix} \in \mathbb{R}^{6\times n}$ has rank 6:

$$\text{rank} \begin{bmatrix} m^{-1}\mathbf{RF} \\ \mathbf{J}^{-1}\mathbf{M} \end{bmatrix} = \text{rank} \left(\begin{bmatrix} m^{-1}\mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \right) = \text{rank} \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \tag{1.248}$$

and it can be seen that the quadrotor (coplanar or tilted) is in any case underactuated ($n = 4$) and the hexarotor can be fully actuated only if $\mathbf{F}$ has the first two rows that are not fully null, hence it needs a tilted configuration of the rotors (opportunely tilted).

Example 8 *Consider the following cases:*

Coplanar quadcopter *: all propellers z-axes are aligned with the body frame z-axis*

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ * & * & * & * \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0 & * & 0 & * \\ * & 0 & * & 0 \\ * & * & * & * \end{bmatrix}$$

$$\textit{rank}(\mathbf{F}) = 1 \qquad \textit{rank}(\mathbf{M}) = 3 \qquad \textit{rank}\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = 4$$

α-tilted quadcopter *: all propellers z-axes are tilted by an angle α around the arm-axis*

$$\mathbf{F} = \begin{bmatrix} 0 & * & 0 & * \\ * & 0 & * & 0 \\ * & * & * & * \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0 & * & 0 & * \\ * & 0 & * & 0 \\ * & * & * & * \end{bmatrix}$$

$$\textit{rank}(\mathbf{F}) = 3 \qquad \textit{rank}(\mathbf{M}) = 3 \qquad \textit{rank}\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = 4$$

β-tilted quadcopter *: all propellers z-axes are tilted by an angle β around the orthogonal to the arm-axis*

$$\mathbf{F} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}$$

$$\textit{rank}(\mathbf{F}) = 3 \qquad \textit{rank}(\mathbf{M}) = 3 \qquad \textit{rank}\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = 4$$

Coplanar hexacopter *: all propellers z-axes are aligned with the body frame z-axis*

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & * & * \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0 & * & * & 0 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}$$

$$\textit{rank}(\mathbf{F}) = 1 \qquad \textit{rank}(\mathbf{M}) = 3 \qquad \textit{rank}\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = 4$$

α-tilted hexacopter *: all propellers z-axes are tilted by an angle α around the arm-axis*

$$\mathbf{F} = \begin{bmatrix} 0 & * & * & 0 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0 & * & * & 0 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}$$

$$\textit{rank}(\mathbf{F}) = 3 \qquad \textit{rank}(\mathbf{M}) = 3 \qquad \textit{rank}\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = 6$$

$\alpha\beta$-tilted pentacopter *: all propellers z-axes are tilted by an angle α around the arm-axis and an angle β around the orthogonal to the arm-axis*

$$\mathbf{F} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

$$\textit{rank}(\mathbf{F}) = 3 \qquad \textit{rank}(\mathbf{M}) = 3 \qquad \textit{rank}\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = 5$$

There is also a further angle that affect the actuation capability of the coplanar multirotor, and this is the angle γ between the arms of the platform; in particular, when the propellers are tilted (for example α-tilted) the change in γ modifies the intensity and direction of the force component coplanar with the propellers. This can be see for example in the plots of Fig. 1.17.

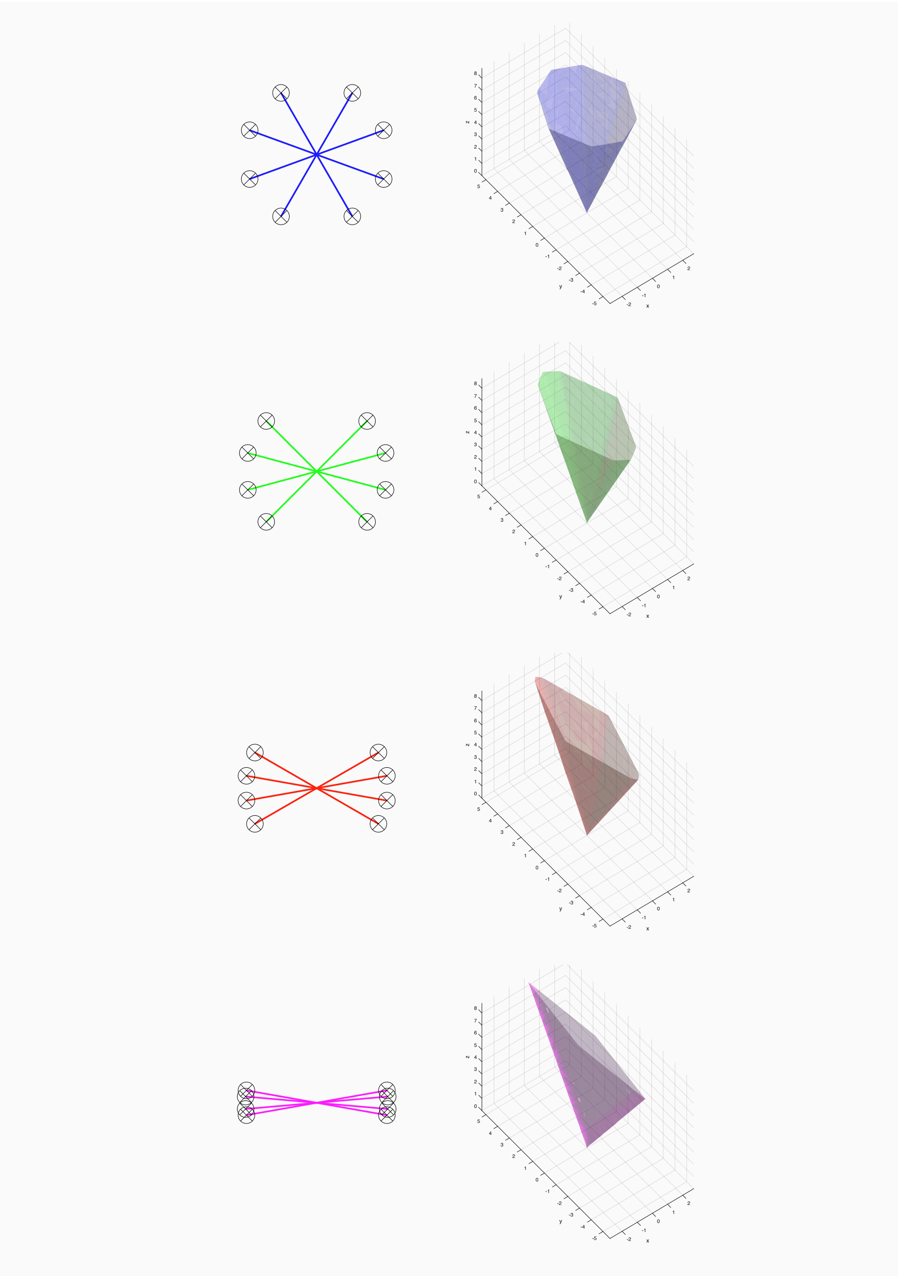


Figure 1.17: Tilted octorotor: force cones of the α-tilted octorotor, for different choice of the γ-angles among the arms.

1.5.3 Actuation and decoupling

In general, for a multirotor platform it stands:

$$\text{control force input matrix} \qquad \mathbf{F} \in \mathbb{R}^{3\times n} \qquad 1 \le \text{rank}(\mathbf{F}) \le 3 \tag{1.249}$$
$$\text{control moment input matrix} \qquad \mathbf{M} \in \mathbb{R}^{3\times n} \qquad 1 \le \text{rank}(\mathbf{M}) \le 3 \tag{1.250}$$

and for the usually considered platforms we have full orientation actuation: rank$(\mathbf{M}) = 3$. Given this premise, it is possible to find matrices $\mathbf{A}$ and $\mathbf{B}$ such that

$$\mathbf{A} \in \mathbb{R}^{n\times 3} \qquad \text{Im}(\mathbf{A}) = \text{Im}(\mathbf{M}^\top) = (\text{ker}(\mathbf{M}))^\perp \tag{1.251}$$
$$\mathbf{B} \in \mathbb{R}^{n\times (n-3)} \qquad \text{Im}(\mathbf{B}) = ker(\mathbf{M}) \tag{1.252}$$

It also follows that $\mathbf{T} = [\mathbf{A}|\mathbf{B}] \in \mathbb{R}^{n\times n}$ is full rank and hence it can be interpreted as a matrix of change of basis in $\mathbb{R}^n$, which can be employed to the input vector $\mathbf{u}$ as:

$$\mathbf{u} = \mathbf{T}\tilde{\mathbf{u}} = [\mathbf{A}|\mathbf{B}] \begin{bmatrix} \tilde{\mathbf{u}}_A \\ \tilde{\mathbf{u}}_B \end{bmatrix} = \mathbf{A}\tilde{\mathbf{u}}_\mathbf{A} + \mathbf{B}\tilde{\mathbf{u}}_\mathbf{B} \tag{1.253}$$

This decomposition of the input vector leads to

$$\mathbf{f}_c = \mathbf{F}\mathbf{u} = \mathbf{F}\mathbf{A}\tilde{\mathbf{u}}_A + \mathbf{F}\mathbf{B}\tilde{\mathbf{u}}_B =: \mathbf{f}_c^A + \mathbf{f}_c^B \tag{1.254}$$
$$\tau_c = \mathbf{M}\mathbf{u} = \mathbf{M}\mathbf{A}\tilde{\mathbf{u}}_A + \mathbf{M}\mathbf{B}\tilde{\mathbf{u}}_B =: \boldsymbol{\tau}_c^A \tag{1.255}$$

and the following spaces can be defined:

$$\begin{aligned}
\mathbf{f}_c &\in \mathcal{F} := \text{Im}(\mathbf{F}) \subseteq \mathbb{R}^3 \\
\mathbf{f}_c^A &\in \mathcal{F}_A := \text{Im}(\mathbf{F}\mathbf{A}) \subseteq \mathcal{F} \\
\mathbf{f}_c^B &\in \mathcal{F}_B := \text{Im}(\mathbf{F}\mathbf{B}) \subseteq \mathcal{F}
\end{aligned}$$

Hence:

- if $\dim(\mathcal{F}_B) = \dim(\mathcal{F})$, it means that control force is completely independent on control moment and the system is *uncoupled*;
- if $0 < \dim(\mathcal{F}_B) < \dim(\mathcal{F})$, at least a component of control force can be chosen freely and the system is *partially coupled*;
- if $0 = \dim(\mathcal{F}_B)$, control force is completely dependent on control moment and the system is *fully coupled*.

Example 9 *Consider the following numerical cases:*

Coplanar quadcopter *: all propellers z-axes are aligned with the body frame z-axis*

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0 & 0.02 & 0 & -0.02 \\ -0.02 & 0 & 0.02 & 0 \\ -0.1 & 0.1 & -0.1 & 0.1 \end{bmatrix}$$

$$\textit{Im}(\mathbf{M}^\top) = < \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} > \qquad \textit{ker}(\mathbf{M}) = < \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} >$$

$$\mathbf{FA} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{FB} = \begin{bmatrix} 0 \\ 0 \\ 0.4 \end{bmatrix} \quad \Rightarrow \quad \begin{cases} \dim(\mathcal{F}_B) = \dim(\mathcal{F}) : \textit{Uncoupled} \\ \mathbf{f}_c = \mathbf{Fu} = \mathbf{FA}\tilde{\mathbf{u}}_A + \mathbf{FB}\tilde{\mathbf{u}}_B = \mathbf{f}_c^B \end{cases}$$

A more detailed analysis of the decomposition of the force vector yields:

$$\mathbf{u} = \mathbf{A}\tilde{\mathbf{u}}_A + \mathbf{B}\tilde{\mathbf{u}}_B = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} d = \begin{bmatrix} -b-c+d \\ a+c+d \\ b-c+d \\ -a+c+d \end{bmatrix}$$

and then

$$\mathbf{f}_c = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix} \begin{bmatrix} -b-c+d \\ a+c+d \\ b-c+d \\ -a+c+d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.4d \end{bmatrix}$$

α-tilted quadcopter *: all propellers z-axes are tilted by an angle α around the arm-axis*

$$\mathbf{F} = \begin{bmatrix} 0 & -0.02 & 0 & 0.02 \\ -0.02 & 0 & 0.02 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0 & 0.01 & 0 & -0.01 \\ -0.01 & 0 & 0.01 & 0 \\ -0.1 & 0.1 & -0.1 & 0.1 \end{bmatrix}$$

$$\textit{Im}(\mathbf{M}^\top) = < \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} > \qquad \textit{ker}(\mathbf{M}) = < \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} >$$

$$\mathbf{FA} = \begin{bmatrix} -0.03 & 0 & 0 \\ 0 & 0.03 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{FB} = \begin{bmatrix} 0 \\ 0 \\ 0.4 \end{bmatrix} \quad \Rightarrow \quad \begin{cases} 0 < \dim(\mathcal{F}_B) < \dim(\mathcal{F}) : \textit{Partially coupled} \\ \mathbf{f}_c = \mathbf{Fu} = \mathbf{FA}\tilde{\mathbf{u}}_A + \mathbf{FB}\tilde{\mathbf{u}}_B = \mathbf{f}_c^A + \mathbf{f}_c^B \end{cases}$$

α**-tilted hexacopter** *: all propellers z-axes are tilted by an angle* α *around the arm-axis*

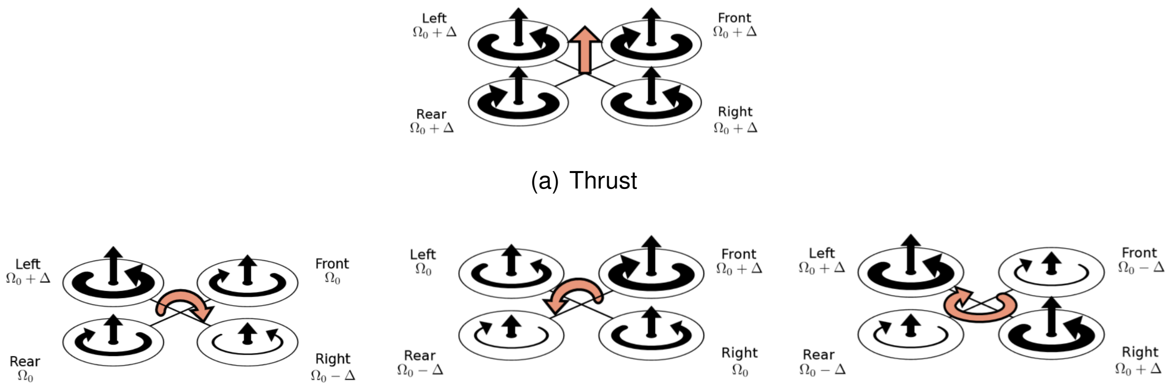
$$\mathbf{F} = \begin{bmatrix} 0 & -0.06 & 0.06 & 0 & -0.06 & 0.06 \\ -0.07 & 0.03 & 0.03 & -0.07 & 0.03 & 0.03 \\ 0.07 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0 & -0.05 & -0.05 & 0 & 0.05 & 0.05 \\ 0.06 & 0.03 & -0.03 & -0.06 & -0.03 & 0.03 \\ -0.08 & 0.08 & -0.08 & 0.08 & -0.08 & 0.08 \end{bmatrix}$$

$$\textit{Im}(\mathbf{M}^\top) =< \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} > \qquad \textit{ker}(\mathbf{M}) =< \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} >$$

$$\mathbf{FA} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{FB} = \begin{bmatrix} 0 & -0.12 & 0.12 \\ -0.14 & 0.07 & 0.07 \\ 0.14 & 0.14 & 0.14 \end{bmatrix} \qquad \Rightarrow \quad \begin{cases} \dim(\mathcal{F}_B) = \dim(\mathcal{F}) : \textit{Uncoupled} \\ \mathbf{f}_c = \mathbf{F}\mathbf{u} = \mathbf{FA}\tilde{\mathbf{u}}_A + \mathbf{FB}\tilde{\mathbf{u}}_B = \mathbf{f}_c^B \end{cases}$$

1.5.4 Platform control

An example of the basic control actions for a quadcopter to obtain thrust and yield roll-pitch-yaw movements is given in Fig. 1.18.



(a) Thrust

(b) Roll (c) Pitch (d) Yaw

Figure 1.18: Basic control actions for a quadcopter: Ω_0 indicates the hovering velocity for the rotors, $\omega_i = \Omega_0$, and Δ the variation to obtain the specific actions.

A full control scheme is given in Fig. 1.19 where the idea of cascaded control is shown: the controller is partitioned into two main blocks, one related to position control and the other related to attitude control. The former takes a position reference $\mathbf{p}_{ref}$ as the input together with the feedback position (and velocity) signal; the latter considers the information related to the body frame orientation (reference rotations, feedback signals). More importantly, the position controller may produce a desired rotation to reach the position reference, which has to be aligned with the orientation reference inside the attitude controller.
These two blocks produce the wrench pair (force and moment), which is translated through the wrench mapper into the actual n-rotor commands (rotational speed of the propellers). A note about an additional feedback acting on the position controller is due: the vertical force to compensate the gravity may change according to the platform orientation, as, for example, in the case of the quadrotor. For this reason, a solution could be to take into account (possibly non-zero) the roll and pitch angles, in order to get exact vertical compensation.

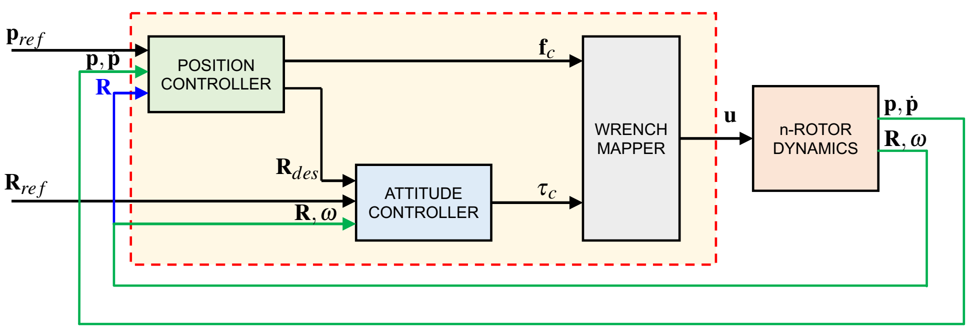


Figure 1.19: Cascaded control scheme: feedback signals are shown in green lines, while the blue connection highlight the possible attitude feedback entering the position controller.

In the case of the quadrotor, the control degrees of freedom are limited to four, because of the structural underactuation of the platform. We can consider, for example, the three positions and the yaw angle (heading direction), while roll and pitch are given respectively by the position y-x controller. Again, also in this scheme we can appreciate through this link the coupling between the translational and rotational dynamics.
The control scheme in this case is shown in Fig. 1.20 and can be realized by a set of standard SISO controllers (for example PID controllers), one for each quantity.

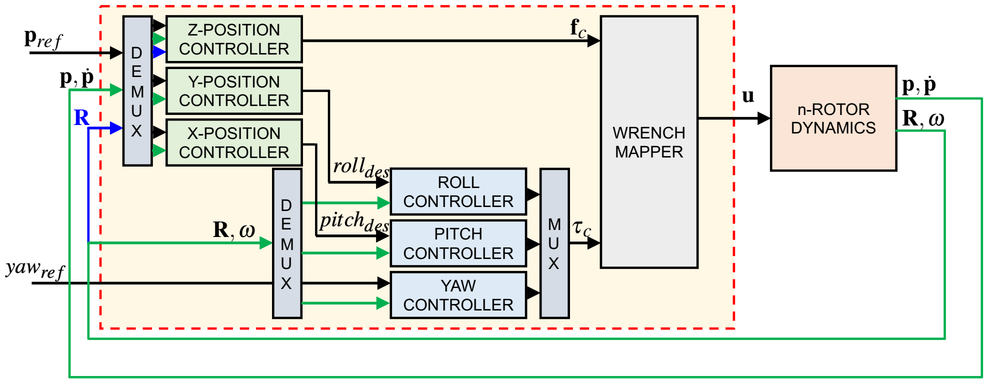


Figure 1.20: Cascaded control scheme for the quadrotor.

Decoupled control

We start from the (1.242) and (1.184) and we consider a diagonal inertia matrix

$$\mathbf{J} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \tag{1.256}$$

and the control inputs given by thrust $\mathbf{f}_c$ and torque $\boldsymbol{\tau}_c$ in body frame

$$\mathbf{f}_c = \mathbf{F}\mathbf{u} = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \qquad \boldsymbol{\tau}_c = \mathbf{M}\mathbf{u} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}; \tag{1.257}$$

the model results as

$$\dot{\mathbf{p}} = \mathbf{v} \tag{1.258}$$

$$\begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix}^{-1} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{1.259}$$

$$\dot v = \ddot{\mathbf{p}} = -\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m}\begin{bmatrix} * & * & s\psi s\phi + c\psi s\theta c\phi \\ * & * & -c\psi s\phi + s\psi s\theta c\phi \\ * & * & c\theta c\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{1.260}$$

$$\begin{bmatrix} \dot\omega_x \\ \dot\omega_y \\ \dot\omega_z \end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x}\omega_y\omega_z \\ \frac{J_z - J_x}{J_y}\omega_z\omega_x \\ \frac{J_x - J_y}{J_z}\omega_x\omega_y \end{bmatrix} + \begin{bmatrix} J_x^{-1} & 0 & 0 \\ 0 & J_y^{-1} & 0 \\ 0 & 0 & J_z^{-1} \end{bmatrix} \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{1.261}$$

and with the additional assumptions of small angles and similar values for for the inertia moments, this further simplifies to

$$\dot{\mathbf{p}} = \mathbf{v} \tag{1.262}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{1.263}$$

$$\ddot{\mathbf{p}} = \dot{v} = -\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \begin{bmatrix} s\psi s\phi + c\psi s\theta c\phi \\ -c\psi s\phi + s\psi s\theta c\phi \\ c\theta c\phi \end{bmatrix} \frac{T}{m} \tag{1.264}$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \tau_\phi / J_x \\ \tau_\theta / J_y \\ \tau_\psi / J_z \end{bmatrix} \tag{1.265}$$

This simplified model can be used to design the control action, where we observe that the inputs thrust T and torques $\tau_\phi, \tau_\theta, \tau_\psi$ are the four control handles given by the quadrotor actuation properties.

Feedback Control #1: Attitude + Elevation Controller

A first choice for the controller references can now be made by considering two loops, one for the attitude (thus requiring a desired attitude profile $\phi^{\text{des}}(t)$, $\theta^{\text{des}}(t)$, $\psi^{\text{des}}(t)$) and a second one for the elevation (thus requiring a desired elevation profile $p_z^{\text{des}}(t)$).
The attitude controller may consist of three independent PDs

$$\tau_\phi = k_{\phi,p}\left(\phi^{\text{des}} - \phi\right) + k_{\phi,d}\left(\dot{\phi}^{\text{des}} - \dot{\phi}\right) \tag{1.266}$$

$$\tau_\theta = k_{\theta,p}\left(\theta^{\text{des}} - \theta\right) + k_{\theta,d}\left(\dot{\theta}^{\text{des}} - \dot{\theta}\right) \tag{1.267}$$

$$\tau_\psi = k_{\psi,p}\left(\psi^{\text{des}} - \psi\right) + k_{\psi,d}\left(\dot{\psi}^{\text{des}} - \dot{\psi}\right) \tag{1.268}$$

while for the elevation controller we may choose an extremely basic action derived from the model equations when the hovering condition is considered ($\phi \approx 0$, $\theta \approx 0$):

$$\ddot{p}_z \approx -g + \frac{T}{m} \Rightarrow T = m\left(\ddot{p}_z^{\text{des}} + g\right) \tag{1.269}$$

or a more refined PD controller that exploits information on the roll ϕ and pitch θ angles, with the nonlinear compensation term for the gravity and the roll and pitch inclinations, and a possible feedforward term

$$T = \frac{m}{c\theta\ c\phi}\left[g + \ddot{p_z}^{\text{des}} + k_{z,p}\left(p_z^{\text{des}} - p_z\right) + k_{z,d}\left(\dot{p_z}^{\text{des}} - \dot{p_z}\right)\right] \tag{1.270}$$

The main disadvantage of this control design is the difficulty of inferring the references for the attitude, given a desired trajectory to follow.

• Error dynamics

$\rightsquigarrow \boxed{e_\alpha = \alpha - \alpha_d}$

$J\ddot{\alpha} = \tau \rightsquigarrow \alpha$ dynamics

$\stackrel{!}{=} k_p(\alpha_d - \alpha) + k_d(\dot{\alpha}_d - \dot{\alpha})$

with $\ddot{\alpha}_d = 0$ we have

$$\boxed{J\,\ddot{e}_\alpha + k_d\,\dot{e}_\alpha + k_p\,e_\alpha = 0}$$

$e_\alpha \longrightarrow 0$

$k_d > 0 \quad k_p > 0$

$\rightsquigarrow \boxed{e_z = z - z_d}$

$\ddot{z} = -g + (c_\theta c_\varphi) T/m \rightsquigarrow z$ dyn

$\stackrel{!}{=} -g + g + \ddot{z}_d + k_{zp}(z_d - z) + k_{zd}(\dot{z}_d - \dot{z})$

we have

$$\boxed{\ddot{e}_z + k_{zd}\,\dot{e}_z + k_{zp}\,e_z = 0}$$

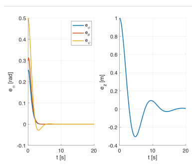
$e_z \longrightarrow 0$

$k_{zd} > 0 \quad k_{zp} > 0$

$\Rightarrow$ 2(x3) gains for the α control
2 gains for the z control

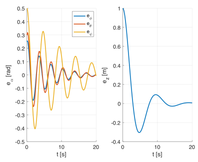
$k_{p\alpha} = 0.1 \quad k_{d\alpha} = 0.1$

$k_{pz} = 0.5 \quad k_{dz} = 0.5$



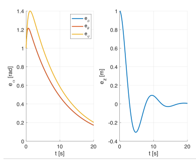
$k_{p\alpha} = 0.1 \quad k_{d\alpha} = 0.01$

$k_{pz} = 0.5 \quad k_{dz} = 0.5$

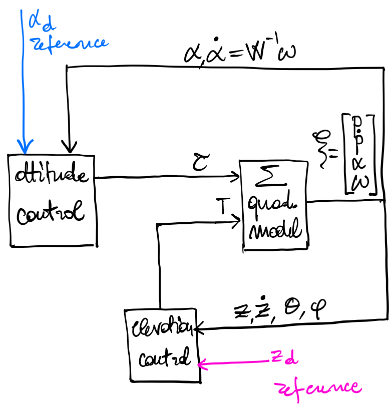


$k_{p\alpha} = 0.01 \quad k_{d\alpha} = 0.1$

$k_{pz} = 0.5 \quad k_{dz} = 0.5$



Control scheme



Feedback Control #2: Yaw angle + Position Controller

This strategy aims at following a desired trajectory specified by the desired positions $\mathbf{p}_T(t)$ and the desired yaw angle $\psi_T(t)$.
We define the position error as $\mathbf{e} = \mathbf{p}_T - \mathbf{p}$ and we compute the desired linear acceleration $\ddot{\mathbf{p}}^{\text{des}}$ using a PID controller

$$\ddot{\mathbf{p}}^{\text{des}} = \ddot{\mathbf{p}}_T + k_d\dot{\mathbf{e}} + k_p\mathbf{e} + k_i\int\mathbf{e} \tag{1.271}$$

To translate $\ddot{\mathbf{p}}^{\text{des}}$ into ϕ^{des} and θ^{des} we consider the linearization of the position dynamics and the given ψ_T

$$\begin{aligned}
\ddot{\mathbf{p}}_x &= (s\psi s\phi + c\psi s\theta c\phi)\frac{T}{m} \approx \left(s\psi_T\phi^{\text{des}} + c\psi_T\theta^{\text{des}}\right)\frac{T^{\text{des}}}{m} && (1.272)\\
\ddot{\mathbf{p}}_y &= (-c\psi s\phi + s\psi s\theta c\phi)\frac{T}{m} \approx \left(-c\psi_T\phi^{\text{des}} + s\psi_T\theta^{\text{des}}\right)\frac{T^{\text{des}}}{m} && (1.273)\\
\ddot{\mathbf{p}}_z &= -g + (c\theta c\phi)\frac{T}{m} \approx -g + \frac{T^{\text{des}}}{m} && (1.274)
\end{aligned}$$

and T^{des} can be computed from the third equation, considering the hovering condition, as $T^{\text{des}} \approx mg$. It follows

$$\ddot{\mathbf{p}}_x \approx g\left(s\psi_T\phi^{\text{des}} + c\psi_T\theta^{\text{des}}\right) \tag{1.275}$$

$$\ddot{\mathbf{p}}_y \approx g\left(-c\psi_T\phi^{\text{des}} + s\psi_T\theta^{\text{des}}\right) \tag{1.276}$$

which can be written in matricial form and inverted (always...) in order to provide the needed attitude references

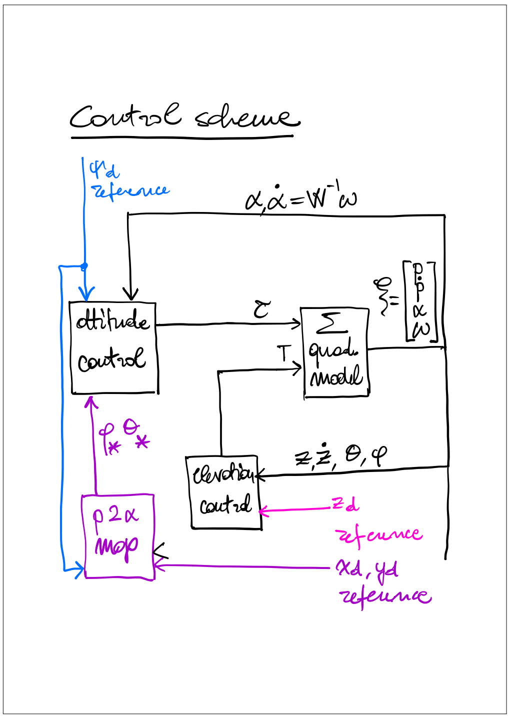
$$\begin{aligned}
\phi^{\text{des}} &= \frac{1}{g}\left(\ddot{p_x}^{\text{des}}s\psi_T - \ddot{p_y}^{\text{des}}c\psi_T\right) && (1.277)\\
\theta^{\text{des}} &= \frac{1}{g}\left(\ddot{p_x}^{\text{des}}c\psi_T + \ddot{p_y}^{\text{des}}s\psi_T\right) && (1.278)
\end{aligned}$$

These are passed to the attitude controller to produce the required torque, to be passed to the wrench mapper

$$\begin{aligned}
\tau_\phi &= k_{\phi,p}\left(\phi^{\text{des}} - \phi\right) + k_{\phi,d}\left(\dot{\phi}^{\text{des}} - \dot{\phi}\right) && (1.279)\\
\tau_\theta &= k_{\theta,p}\left(\theta^{\text{des}} - \theta\right) + k_{\theta,d}\left(\dot{\theta}^{\text{des}} - \dot{\theta}\right) && (1.280)\\
\tau_\psi &= k_{\psi,p}\left(\psi_T - \psi\right) + k_{\psi,d}\left(\dot{\psi}_T - \dot{\psi}\right) && (1.281)
\end{aligned}$$

In addition, and similarly to what done before, the value of the thrust T is passed to the wrench mapper as

$$T = m\left(\ddot{p}_z^{\text{des}} + g\right). \tag{1.282}$$



Control scheme
φ_d reference
$\alpha, \dot{\alpha} = W^{-1}\omega$
attitude control
τ
T
Σ quad. model
$\xi = [p, \dot{p}, \alpha, \omega]$
φ_* θ_*
$z, \dot{z}, \theta, \varphi$
elevation control
z_d reference
p2α map
x_d, y_d reference

Feedback Control #3: Output Feedback Control with Feedback Linearization

Dynamic Feedback Linearization

$\Rightarrow$ Simplified model for control design

$$\boxed{\begin{aligned}\ddot{p} &= \begin{bmatrix} (s\varphi s\psi + c\varphi s\theta c\psi)\, T/m \\ (-c\varphi s\psi + s\varphi s\theta c\psi)\, T/m \\ -g + (c\theta c\varphi)\, T/m \end{bmatrix} \\ \begin{bmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} &= \begin{bmatrix} \tau_\varphi / J_x \\ \tau_\theta / J_y \\ \tau_\psi / J_z \end{bmatrix}\end{aligned}}$$

Second order model with

- state: $\xi = [p \;\; \dot{p} \;\; [\varphi \;\theta\; \psi] \;\; \omega]^T$
- input: $u = [T \;\; \tau_\varphi \;\; \tau_\theta \;\; \tau_\psi]^T$

$$\underbrace{\begin{bmatrix} \ddot{p} \\ \hline \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}}_{\dot{\xi}} = \underbrace{\begin{bmatrix} -g \\ \hline \\ \\ \end{bmatrix}}_{f(\xi)} + \underbrace{\begin{bmatrix} (s\varphi s\psi + c\varphi s\theta c\psi)\, T/m \\ (-c\varphi s\psi + s\varphi s\theta c\psi)\, T/m \\ (c\theta c\varphi)\, T/m \\ \hline \tau_\varphi / J_x \\ \tau_\theta / J_y \\ \tau_\psi / J_z \end{bmatrix}}_{g(\xi)\,u}$$

$$\dot{\xi} = f(\xi) + g(\xi)\, u$$

$\xi \in \mathbb{R}^6$, $u \in \mathbb{R}^4$ $\Rightarrow$ underactuation

Note

Given the underactuation the system cannot be transformed into an equivalent linear controllable system via static state feedback

$\Downarrow$

Full state linearization can be obtained via dynamic state feedback

- Define the output to be controlled

$\eta(\xi) = [p \,,\, \psi]^T \rightarrow$ position yaw ang

$\Downarrow$

output dynamics

$$\begin{bmatrix} \boxed{\ddot{p}} \\ \hline \ddot{\varphi} \\ \ddot{\theta} \\ \boxed{\ddot{\psi}} \end{bmatrix} = \begin{bmatrix} -g \\ \hline \\ \\ \end{bmatrix} + \begin{bmatrix} (s\varphi s\psi + c\varphi s\theta c\psi)\, T/m \\ (-c\varphi s\psi + s\varphi s\theta c\psi)\, T/m \\ (c\theta c\varphi)\, T/m \\ \hline \tau_\varphi / J_x \\ \tau_\theta / J_y \\ \tau_\psi / J_z \end{bmatrix}$$

• Idea in a nutshell...

$$\left[\begin{array}{c}\ddot{p}\\ \hline \ddot{\psi}\end{array}\right]=\left[\begin{array}{c}0\\0\\-g\\ \hline 0\end{array}\right]+\left[\begin{array}{c|ccc}(s\varphi s\psi+c\varphi s\theta c\psi)/m & 0&0&0\\(-c\varphi s\psi+s\varphi s\theta c\psi)/m & 0&0&0\\(c\theta c\varphi)/m & 0&0&0\\ \hline 0 & 0&0&J_z^{-1}\end{array}\right]\left[\begin{array}{c}T\\ \hline \tau_\varphi\\ \tau_\theta\\ \tau_\psi\end{array}\right]$$

$$\left|\begin{bmatrix}* & 0 & 0 & 0\\ * & 0 & 0 & 0\\ * & 0 & 0 & 0\\ 0 & 0 & 0 & *\end{bmatrix}\right| = 0 \rightsquigarrow \text{singular!}$$

$$\Downarrow$$

$$\underbrace{\left[\begin{array}{c}\ddddot{p}\\ \hline \ddot{\psi}\end{array}\right]}=\underbrace{\left[\begin{array}{c}\tilde{f}_x\\ \tilde{f}_y\\ \tilde{f}_z\\ \hline 0\end{array}\right]}+\underbrace{\left[\begin{array}{cccc}\tilde{g}_{x1}&\tilde{g}_{x2}&\tilde{g}_{x3}&\tilde{g}_{x4}\\ \tilde{g}_{y1}&\tilde{g}_{y2}&\tilde{g}_{y3}&\tilde{g}_{y4}\\ \tilde{g}_{z1}&\tilde{g}_{z2}&\tilde{g}_{z3}&\tilde{g}_{z4}\\ \hline 0&0&0&1\end{array}\right]}\underbrace{\left[\begin{array}{c}\ddot{T}\\ \hline \tau_\varphi\\ \tau_\theta\\ \tau_\psi\end{array}\right]}$$

$|\cdot| \neq 0 \rightsquigarrow$ non singular!

$$\boxed{v = \tilde{f} + \tilde{g}\cdot\tilde{u}}$$

where $v = \left[\begin{array}{c}\ddddot{p}\\ \hline \ddot{\psi}\end{array}\right]$ $\quad \tilde{u} = \left[\begin{array}{c}\ddot{T}\\ \hline \tau\end{array}\right]$

If $\tilde{g}$ non singular $\Rightarrow$ invertible

$$\boxed{\tilde{u} = \tilde{g}^{-1}(v - \tilde{f})}$$

is the linearizing and decoupling control law

$$\Downarrow$$

v is the new control input s.t. the closed loop system results

$$\left[\begin{array}{c}\ddddot{p}\\ \ddot{\psi}\end{array}\right] = v$$

with the new control signal v that we are going to design according to the desired reference value...

• Idea in details

$$\left[\begin{array}{c}\ddot{p}\\ \hline \ddot{\psi}\end{array}\right]=\left[\begin{array}{c}0\\0\\-g\\ \hline 0\end{array}\right]+\left[\begin{array}{c|ccc}(s\varphi s\psi+c\varphi s\theta c\psi)/m & 0&0&0\\(-c\varphi s\psi+s\varphi s\theta c\psi)/m & 0&0&0\\(c\theta c\varphi)/m & 0&0&0\\ \hline 0 & 0&0&J_z^{-1}\end{array}\right]\left[\begin{array}{c}T\\ \hline \tau_\varphi\\ \tau_\theta\\ \tau_\psi\end{array}\right]$$

$$\Downarrow$$

$$\left[\begin{array}{c}m\ddot{p}\\ \hline J_z\ddot{\psi}\end{array}\right]=\left[\begin{array}{c}\begin{bmatrix}0\\0\\-mg\end{bmatrix}\\ \hline 0\end{array}\right]+\left[\begin{array}{c|c}A_{3\times1} & 0_{3\times3}\\ \hline 0 & [0\ 0\ 1]\end{array}\right]\left[\begin{array}{c}T\\ \hline \tau\end{array}\right]$$

call $\alpha = \begin{bmatrix}\varphi\\ \theta\\ \psi\end{bmatrix} \longrightarrow \ddot{\alpha} = \begin{bmatrix}\ddot{\varphi}\\ \ddot{\theta}\\ \ddot{\psi}\end{bmatrix} = \begin{bmatrix}J_x^{-1}\tau_x\\ J_y^{-1}\tau_y\\ J_z^{-1}\tau_z\end{bmatrix}$

Note: between $\ddot{\alpha}$ and τ we have a direct proportionality...

$$\begin{cases} \ddot{p} = f + \frac{1}{m} A T \\ \ddot{\psi} = 0 + [0\,0\,1]\, \ddot{\alpha} \end{cases}$$

$$\Rightarrow \begin{bmatrix} \ddot{p} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} + \left[\begin{array}{c|c} \frac{1}{m} A & 0 \\ \hline 0 & [0\,0\,1] \end{array}\right] \begin{bmatrix} T \\ \ddot{\alpha} \end{bmatrix}$$

$\rightsquigarrow$ control $[T \;\; \ddot{\alpha}]^T$ (Note)

$$\rightsquigarrow \left| \left[\begin{array}{c|c} \frac{1}{m} A & 0 \\ \hline 0 & [0\,0\,1] \end{array}\right] \right| = 0 \quad !$$

$\rightsquigarrow$ $\boxed{\ddot{p} \longrightarrow \dddot{p}}$

$$\dddot{p} = 0 + \frac{1}{m}\left(A \dot{T} + \dot{A} T\right) = 0 + \frac{1}{m}\left(A \dot{T} + B \dot{\alpha} T\right)$$

$$\Rightarrow \begin{bmatrix} \dddot{p} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{m} B \dot{\alpha} T \\ 0 \end{bmatrix} + \left[\begin{array}{c|c} \frac{1}{m} A & 0 \\ \hline 0 & [0\,0\,1] \end{array}\right] \begin{bmatrix} \dot{T} \\ \ddot{\alpha} \end{bmatrix}$$

$\rightsquigarrow$ control $[\dot{T} \;\; \ddot{\alpha}]^T$ (Note)

$$\rightsquigarrow \left| \left[\begin{array}{c|c} \frac{1}{m} A & 0 \\ \hline 0 & [0\,0\,1] \end{array}\right] \right| = 0 \quad !$$

$\rightsquigarrow$ $\boxed{\dddot{p} \longrightarrow \ddddot{p}}$

$$\ddddot{p} = 0 + \frac{1}{m}\left(A \ddot{T} + \underbrace{\dot{A} \dot{T} + \dot{A} \dot{T}} + \ddot{A} T\right) = 0 + \frac{1}{m}\left(A \ddot{T} + 2 B \dot{\alpha} \dot{T} + \dot{B} \dot{\alpha} T + B \ddot{\alpha} T\right)$$

$$\Rightarrow \begin{bmatrix} \ddddot{p} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}(2 B \dot{\alpha} \dot{T} + \dot{B} \dot{\alpha} T) \\ 0 \end{bmatrix} + \left[\begin{array}{c|c} \frac{1}{m} A & \frac{1}{m} B T \\ \hline 0 & [0\,0\,1] \end{array}\right] \begin{bmatrix} \ddot{T} \\ \ddot{\alpha} \end{bmatrix}$$

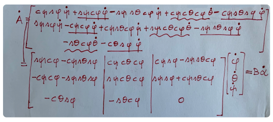
$\rightsquigarrow$ control $[\ddot{T} \;\; \ddot{\alpha}]^T$ (Note)

$$\rightsquigarrow \left| \left[\begin{array}{c|c} \frac{1}{m} A & \frac{1}{m} B T \\ \hline 0 & [0\,0\,1] \end{array}\right] \right| \neq 0 \quad !$$

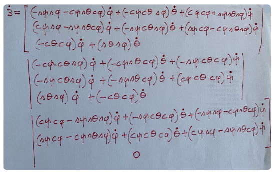
Now...

$\rightsquigarrow$ A is known...

$\rightsquigarrow$ $\dot{A} = B \dot{\alpha}$ is this:



$\rightsquigarrow$ $\ddot{A} = \dot{B} \dot{\alpha} + B \ddot{\alpha}$ so $\dot{B}$ is this:



It follows that – dimensionality check

$$\begin{array}{c} [3] \\ [1] \end{array}\begin{bmatrix}\dddot{p}\\ \hline \ddot{\varphi}\end{bmatrix} = \overbrace{\begin{bmatrix}\frac{1}{m}(2B\dot{\alpha}\dot{T} + \dot{B}\dot{\alpha}T)\\ \hline 0\end{bmatrix}}^{[3\times3][3\times1]+[3\times3][3\times1]} + \overbrace{\begin{bmatrix}\frac{1}{m}A & \frac{1}{m}BT\\ \hline 0 & [0\,0\,1]\end{bmatrix}}^{[3\times1]\,[3\times3]}\begin{bmatrix}\ddot{T}\\ \hline \ddot{\alpha}\end{bmatrix}\begin{array}{c} [1] \\ [3] \end{array}$$

$$\downarrow$$

$$\begin{bmatrix}\dddot{p}\\ \hline \ddot{\varphi}\end{bmatrix} = \begin{bmatrix}*\\ *\\ *\\ \hline 0\end{bmatrix} + \begin{bmatrix}* & * & * & *\\ * & * & * & *\\ * & * & * & 0\\ \hline 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\ddot{T}\\ \hline \ddot{\alpha}\end{bmatrix}$$

- Closed loop system

$$\Rightarrow \begin{bmatrix}\dddot{p}\\ \hline \ddot{\varphi}\end{bmatrix} = \underbrace{\begin{bmatrix}\frac{1}{m}(2B\dot{\alpha}\dot{T} + \dot{B}\dot{\alpha}T)\\ \hline 0\end{bmatrix}}_{\tilde{f}} + \underbrace{\begin{bmatrix}\frac{1}{m}A & \frac{1}{m}BT\\ \hline 0 & [0\,0\,1]\end{bmatrix}}_{\tilde{g}}\begin{bmatrix}\ddot{T}\\ \hline \ddot{\alpha}\end{bmatrix}$$

$$\boxed{\begin{bmatrix}\dddot{p}\\ \ddot{\varphi}\end{bmatrix} = v}$$ closed loop system with decoupled dynamics

with

$$\begin{bmatrix}\ddot{T}\\ \ddot{\alpha}\end{bmatrix} = \tilde{g}^{-1}\left(v - \tilde{f}\right)$$

$v = \begin{bmatrix}v_p\\ v_\varphi\end{bmatrix}$ is the new system input

- Error dynamics

$\rightsquigarrow \boxed{e_\varphi = \varphi - \varphi_d}$

$$v_\varphi = \ddot{\varphi}_d - k_{\varphi_2}(\dot{\varphi} - \dot{\varphi}_d) - k_{\varphi_1}(\varphi - \varphi_d)$$

$$\Downarrow$$

$$\ddot{\varphi} = v_\varphi \Longrightarrow \boxed{\ddot{e}_\varphi + k_{\varphi_1}\dot{e}_\varphi + k_{\varphi_0}e_\varphi = 0}$$

$e_\varphi \longrightarrow 0$

$k_{\varphi_1} > 0 \quad k_{\varphi_0} > 0$

$\rightsquigarrow \boxed{e_p = p - p_d}$

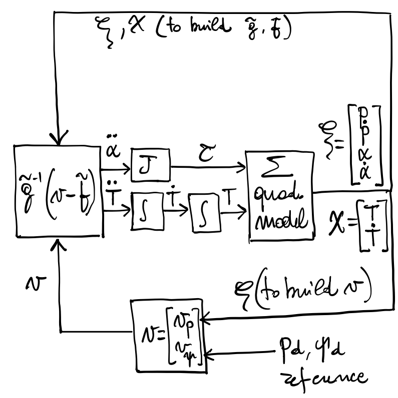
$$v_p = \ddddot{p}_d - k_{p_3}(\dddot{p} - \dddot{p}_d) - k_{p_2}(\ddot{p} - \ddot{p}_d) - k_{p_1}(\dot{p} - \dot{p}_d) - k_{p_0}(p - p$$

$$\Downarrow$$

$$v_p = \ddddot{p} \Longrightarrow \boxed{\ddddot{e}_p + k_{p_3}\dddot{e}_p + k_{p_2}\ddot{e}_p + k_{p_1}\dot{e}_p + k_{p_0}e_p = 0}$$

$\Rightarrow$ 2 gains for the φ control
4(x3) gains for the p control

Control scheme



1.5.5 Quadrotor control demo

Hereafter a standard, planar quadrotor is controlled

- in attitude + elevation, i.e., the goal is to reach a desired attitude (roll, pitch, yaw angles) and a desired altitude, and
- in position, i.e., the goal is to track a given trajectory.

The controllers has been tuned using the MATLAB PID tuner taking into account the saturated outputs.
The complete dynamical model of a quadrotor is given in the previous sections: in particular, the IRIS quadrotor model is considered: it is a slightly asymmetric quadrotor, with almost twice the amount of rolling capabilities compared with the pitching one.
The input $\mathbf{u}$ is the collection of the rotor spinning rates squared $\mathbf{u} = \begin{bmatrix} \omega_1^2 & \omega_2^2 & \omega_3^2 & \omega_4^2 \end{bmatrix}$. To summarize, a wrench mapper can be designed to convert the desired common torque T and the desired body torques $\tau_\phi, \tau_\theta, \tau_\psi$(roll, pitch, yaw, respectively) into the required input vector.

Feedback Control #1: Attitude + Elevation Controller

The attitude controller consists of the three independent PDs

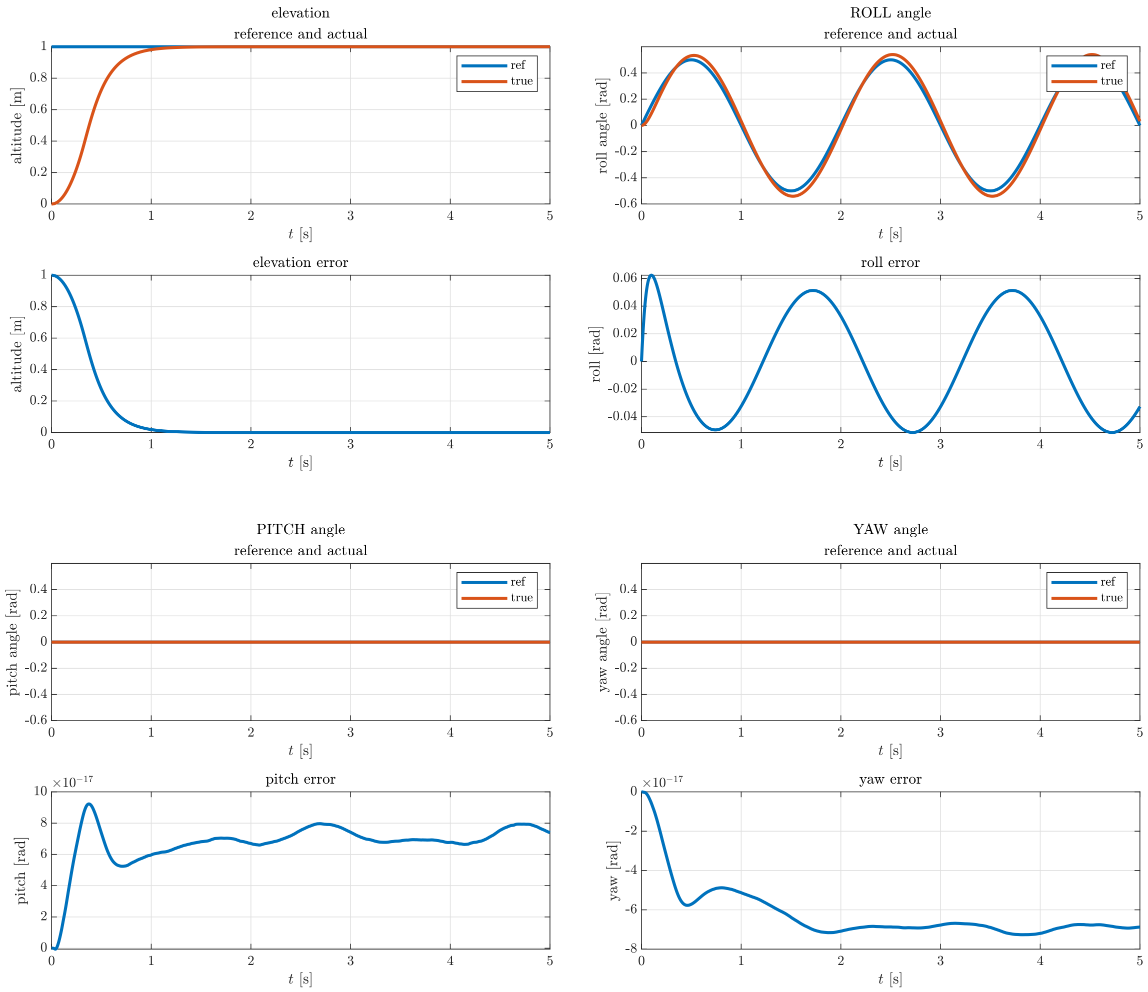
$$\begin{aligned}
\tau_\phi &= k_{\phi,p}\left(\phi^{\text{des}} - \phi\right) + k_{\phi,d}\tfrac{s\,N_\phi}{N_\phi+s}\left(\dot{\phi}^{\text{des}} - \dot{\phi}\right)\\
\tau_\theta &= k_{\theta,p}\left(\theta^{\text{des}} - \theta\right) + k_{\theta,d}\tfrac{s\,N_\theta}{N_\theta+s}\left(\dot{\theta}^{\text{des}} - \dot{\theta}\right)\\
\tau_\psi &= k_{\psi,p}\left(\psi^{\text{des}} - \psi\right) + k_{\psi,d}\tfrac{s\,N_\psi}{N_\psi+s}\left(\dot{\psi}^{\text{des}} - \dot{\psi}\right)
\end{aligned}$$

where N is a filter coefficient needed to implement the derivative term. Moreover, the PDs have saturated outputs.
The elevation controller is also a PD but with the addition of a nonlinear compensation term for the gravity and the roll and pitch inclinations, and a feedforward term

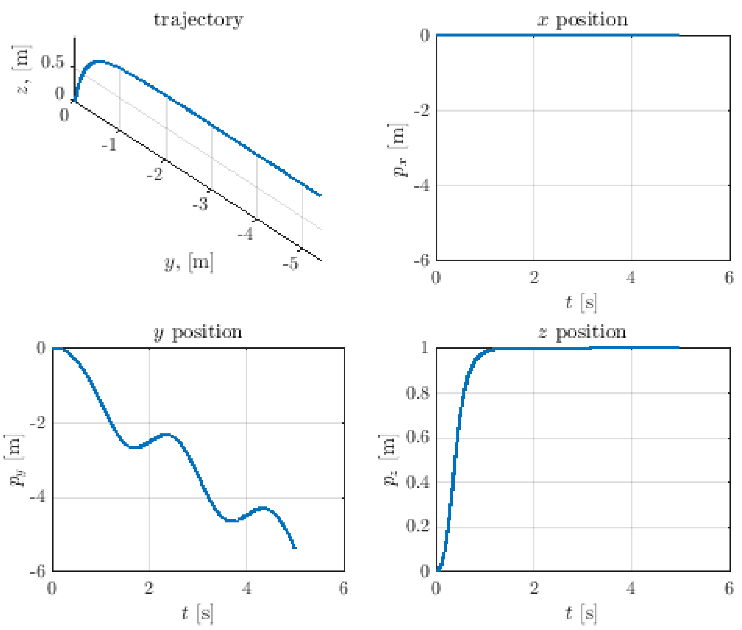
$$T = \frac{m}{c\theta\ c\phi}\left(g + \ddot{p_z}^{\text{des}} + k_{z,p}\left(p_z^{\text{des}} - p_z\right) + k_{z,d}\left(\dot{p_z}^{\text{des}} - \dot{p_z}\right)\right)$$

The controller is evaluated on the following reference signals

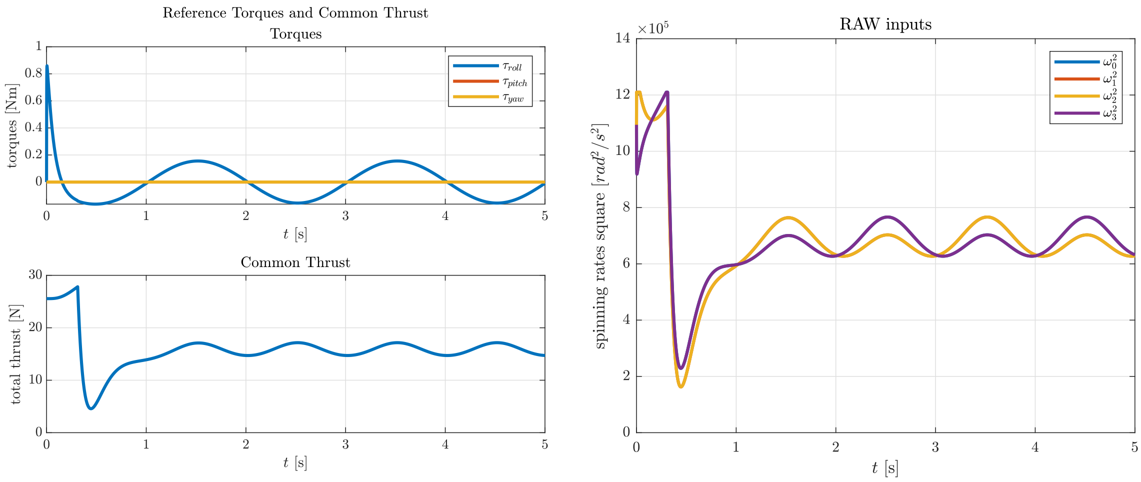
$$\begin{aligned}
&\phi^{\text{ref}}(t) = 0.5\sin(\pi t)\\
&\theta^{\text{ref}}(t) = 0\\
&\psi^{\text{ref}}(t) = 0\\
&{p_z}^{\text{ref}}(t) = 1
\end{aligned}$$



elevation
reference and actual
altitude [m]
ref
true
t [s]
elevation error
altitude [m]
t [s]
ROLL angle
reference and actual
roll angle [rad]
ref
true
t [s]
roll error
roll [rad]
t [s]
PITCH angle
reference and actual
pitch angle [rad]
ref
true
t [s]
pitch error
×10^-17
pitch [rad]
t [s]
YAW angle
reference and actual
yaw angle [rad]
ref
true
t [s]
yaw error
×10^-17
yaw [rad]
t [s]

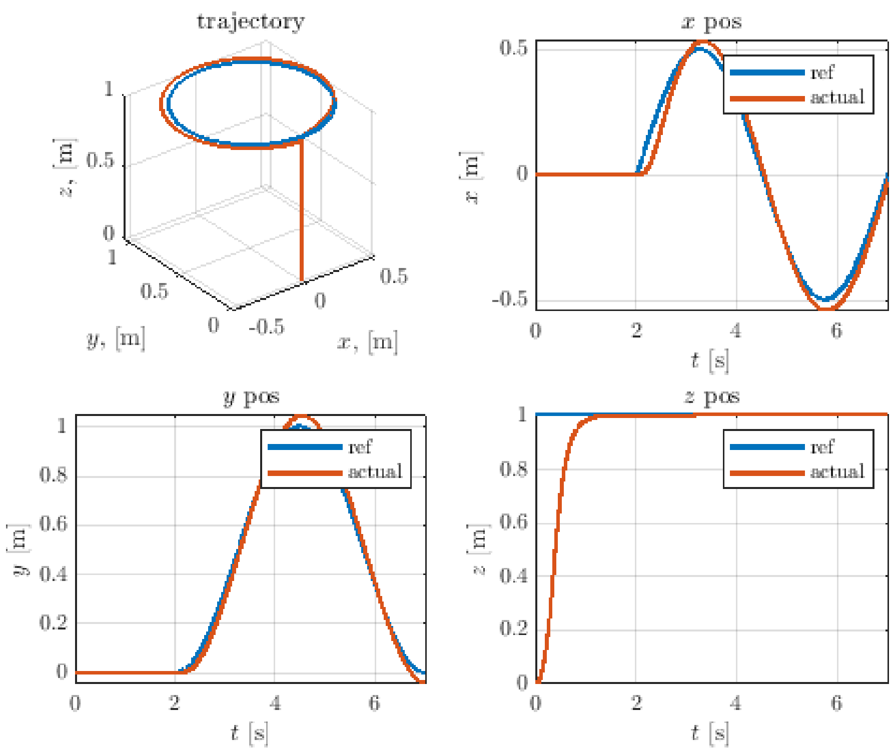


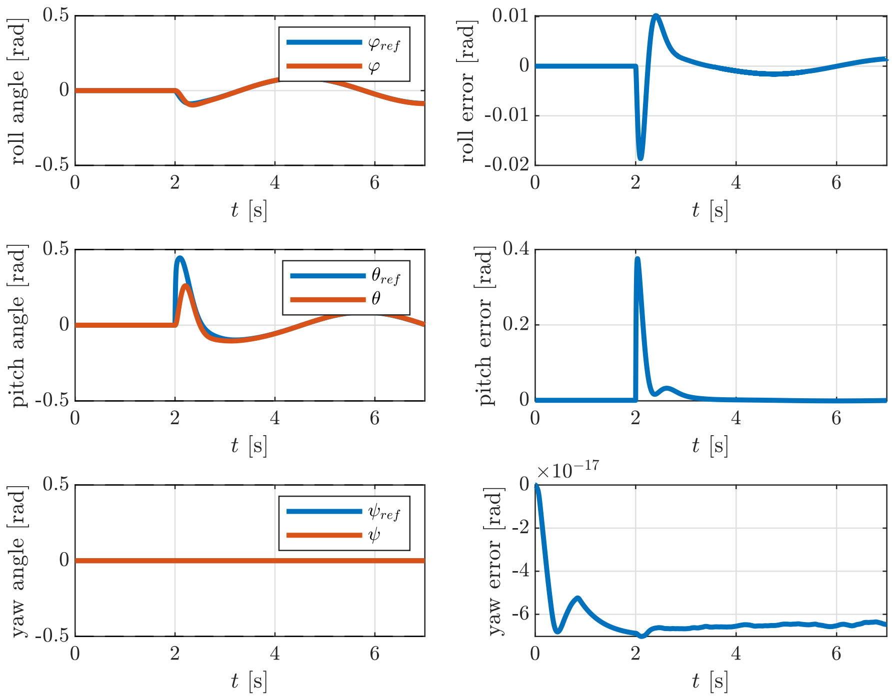
trajectory
z, [m]
y, [m]
x position
p_x [m]
t [s]
y position
p_y [m]
t [s]
z position
p_z [m]
t [s]

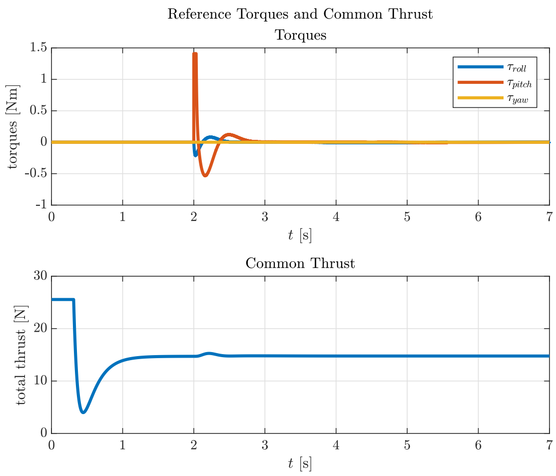


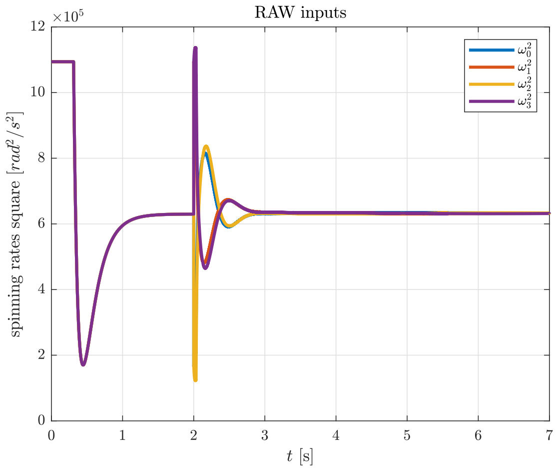
Feedback Control #2: Yaw angle + Position Controller

This controller aims at following a desired trajectory specified by the desired positions $p_T(t)$ and the desired yaw angle $\psi_T(t)$.

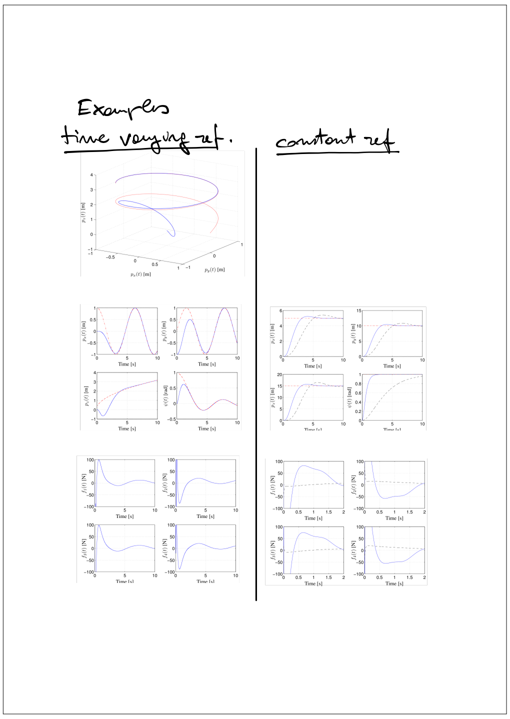



roll angle [rad]
φ_{ref}
φ
roll error [rad]
pitch angle [rad]
θ_{ref}
θ
pitch error [rad]
yaw angle [rad]
ψ_{ref}
ψ
yaw error [rad]
$\times 10^{-17}$
t [s]


Reference Torques and Common Thrust
Torques
torques [Nm]
τ_{roll}
τ_{pitch}
τ_{yaw}
t [s]
Common Thrust
total thrust [N]
t [s]


RAW inputs
$\times 10^5$
spinning rates square $[rad^2/s^2]$
t [s]

Feedback Control #3: Output Feedback Control with Feedback Linearization



1.6 Satellite model

If we consider a satellite model, we can leverage the study performed so far with the aerial vehicles by noting that the Earth gravitational component is not present. Kinematics and dynamics equations read as:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{1.283}$$
$$\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}]_\times \tag{1.284}$$
$$m\ddot{\mathbf{p}} = \mathbf{R}\mathbf{f}_c \tag{1.285}$$
$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau}_c \tag{1.286}$$

These equations have a local validity, meaning that they can describe well scenarios where satellites are in close proximity (for example, in the case of rendez-vous and docking maneuvers).
For more complex behaviors and long-range interactions the above equations need to be coupled with the Clohessy-Whiltshire equations, which model orbital motions (see Fig. 1.21).

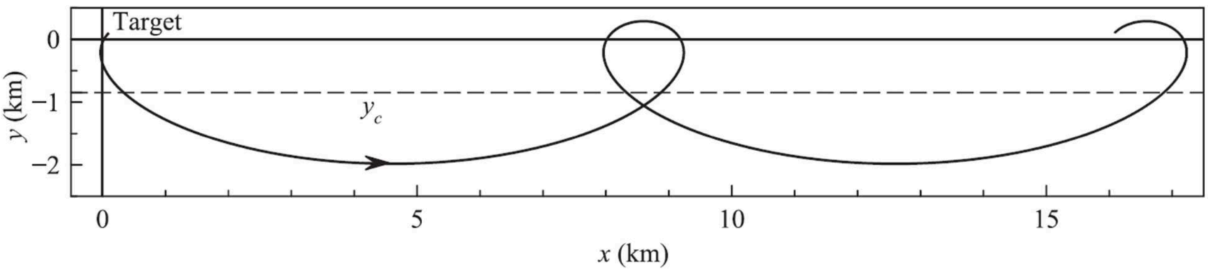


Figure 1.21: From https://arxiv.org/pdf/1908.02592.pdf. *The trajectory of an astronaut attempting to return to her target spacecraft from an initial position of* $x_0 = 100m$ *and* $y_0 = 100m$ *during two orbital periods. Her trajectory carries her away from the target as the center of her drifting elliptical path travels along the line* $y = y_c$.

Chapter 2

Network Theory

2.1 Networked control of multi-agent systems

Given the systems that we have discussed before (e.g. sensors, mobile agents, cameras), we are interested in controlling three different kinds of networks, which are:

- robotic networks, i.e.
  UAVs (Unmanned Aerial Vehicles)
  UGVs (Unmanned Ground Vehicles)
- sensor networks, i.e.
  SANs (Sensor-Actor Networks)
  WSNs (Wireless Sensor Networks)
- camera networks, i.e.
  VSNs (Visual Sensor Networks).

In order to model these networks, we use *Boids models* (Reynolds 1986). The underlying idea is to control the many players encoding three local rules on different spatial scales (form smallest to largest):

- separation $\rightarrow$ avoid collisions with neighbors
- alignment $\rightarrow$ align vector velocity to that of the neighbors
- cohesion $\rightarrow$ avoid dispersion.

These rules give rise to *emergent behaviours* such as *flocking*.

The key ingredients for modelling the MAS are:

- local $\rightarrow$ sensing and information
- exchange $\rightarrow$ communication
- global $\rightarrow$ tasks.

Tasks There are three different types of tasks that we can assign to the agents

- collaborative $\rightarrow$ individual aims that are compatible (e.g.: office work)
- cooperative $\rightarrow$ common objectives that are shared (e.g.: coverage, search and rescue)
- collective $\rightarrow$ common objectives that are not shared (e.g.: flocking).

Communication (Networks) There are three kinds of networks among which we can choose to model the system:

- communication networks, which indicates who can talk with whom
- information networks, which indicates who shares the same information with whom
- geographical networks, which represents the position of the agents inside the environment.

Based on the underlying system we can distinguish three types of models:

- static networks $\rightarrow$ LTI systems (e.g.: fixed cameras)
- dynamic networks $\rightarrow$ hybrid/switching systems (e.g.: drug delivery, gene regulation and brain network)
- random networks $\rightarrow$ stochastic systems (e.g.: imperfect wireless channel).

Protocols of interest There are different objectives that can be pursued:

- Consensus $\rightarrow$ agreement on a state of the system
- Formation $\rightarrow$ controlling position of a robot network in space
- Coverage $\rightarrow$ controlling position of a robot network in space in order to maximize its awareness of the environment
- Assignment $\rightarrow$ distribution of resources and tasks within the network
- Swarming $\rightarrow$ controlling position of very large networks not individually but in average (mean field approach)

2.2 Introduction to the consensus problem

We introduce the problem of consensus with reference to two canonical applications: rendez-vous and distributed estimation. In both cases we end up with an autonomous system of the form

$$\mathbf{x}(k+1) = P\mathbf{x}(k) \tag{2.1}$$

that we want to converge to a unique solution. The main idea behind this algorithm is the following:

1. exchange position information between nodes;
2. update current self-position.

The rendez-vous problem

For example, for the rendez-vous problem, consider the update rule:

$$\begin{cases} x_i(k+1) = \rho_{i1}x_1(k) + \rho_{i2}x_2(k) + ... + \rho_{in}x_n(k) \\ y_i(k+1) = \rho_{i1}y_1(k) + \rho_{i2}y_2(k) + ... + \rho_{in}y_n(k) \end{cases}$$

where $\rho_{ij} \geq 0$ and $\sum_{j=1}^{n} \rho_{ij} = 1$, and $x_i(k)$, $y_i(k)$ is the position of the agent i at time k. Note that ρ_{ij} can be null if the agent i doesn't exchange information with the agent j. Considering just the state x_i in the system, we can notice that

$$\begin{aligned} x_i(k+1) &= \sum_j \rho_{ij}x_j(k) = \rho_{ii}x_i(k) + \sum_{j\neq i} \rho_{ij}x_j(k) \\ &= \left(1 - \sum_{j\neq i} \rho_{ij}\right) x_i(k) + \sum_{j\neq i} \rho_{ij}x_j(k) \\ &= x_i(k) + \underbrace{\sum_{j\neq i} \rho_{ij}(x_j(k) - x_i(k))}_{u_i(k)} \end{aligned}$$

where the first term is the state and the second one is the state dependent input $u_i(k)$. If we consider the problem in matrix notation:

$$\mathbf{x}(k+1) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ * \\ x_n(k) \end{bmatrix} + \underbrace{\begin{bmatrix} (-\rho_{12} - \rho_{13}...) & \rho_{12} & \rho_{13} & * & \rho_{1n} \\ \rho_{21} & (-\rho_{21} - \rho_{23}...) & \rho_{23} & * & \rho_{2n} \\ * & * & * & * & * \\ \rho_{n1} & \rho_{n2} & \rho_{n3} & * & (-\rho_{n1} - \rho_{n2}...) \end{bmatrix}}_{F} \begin{bmatrix} x_1(k) \\ x_2(k) \\ * \\ x_n(k) \end{bmatrix}.$$

From this, we end up having

$$\mathbf{x}(k+1) = \mathbf{x}(k) + F\mathbf{x}(k) = (I_n + F)\mathbf{x}(k) = P\mathbf{x}(k)$$

which is the autonomous system we presented in the beginning, with P provided by

$$P = \begin{bmatrix} (1-\rho_{12}-\rho_{13}...) & \rho_{12} & \rho_{13} & * & \rho_{1n} \\ \rho_{21} & (1-\rho_{21}-\rho_{23}...) & \rho_{23} & * & \rho_{2n} \\ * & * & * & * & * \\ \rho_{n1} & \rho_{n2} & * & * & (1-\rho_{n1}-\rho_{n2}...) \end{bmatrix}.$$

In the forthcoming, let $\mathbb{1}$ denote the vector $\underbrace{\begin{bmatrix}1 \dots 1\end{bmatrix}^\top}_{n \text{ components}}$.

We recognize that P is a stochastic matrix, that means that all its entries are non-negative and all its rows sum to 1. Then, since P is stochastic, it follows

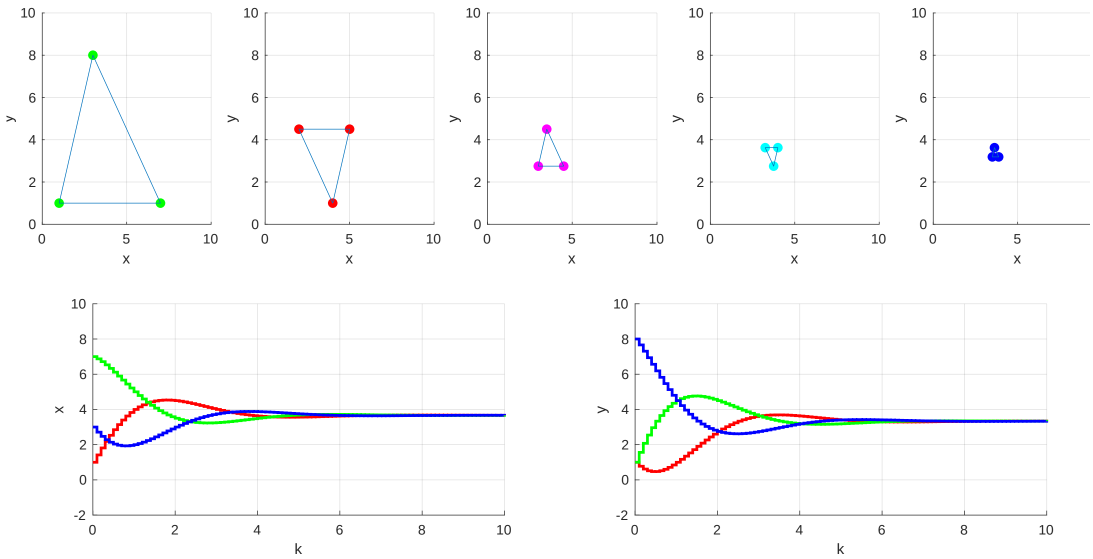
$$P\mathbb{1} = 1\mathbb{1}$$

which means that $\mathbb{1}$ is the right eigenvector relative to the eigenvalue 1. In this case, we would like that the consensus problem converges to a unique solution $\bar{P}x(0) = \alpha\mathbb{1}$, meaning that all the agents converge to the rendez-vous position (in the plane).

Here, it follows an example of planar rendez-vous with three agents that follow update law (2.1) with different P (both the matrix P, its spectrum $\sigma(P)$, and the convergence matrix $\bar{P}$ are given):

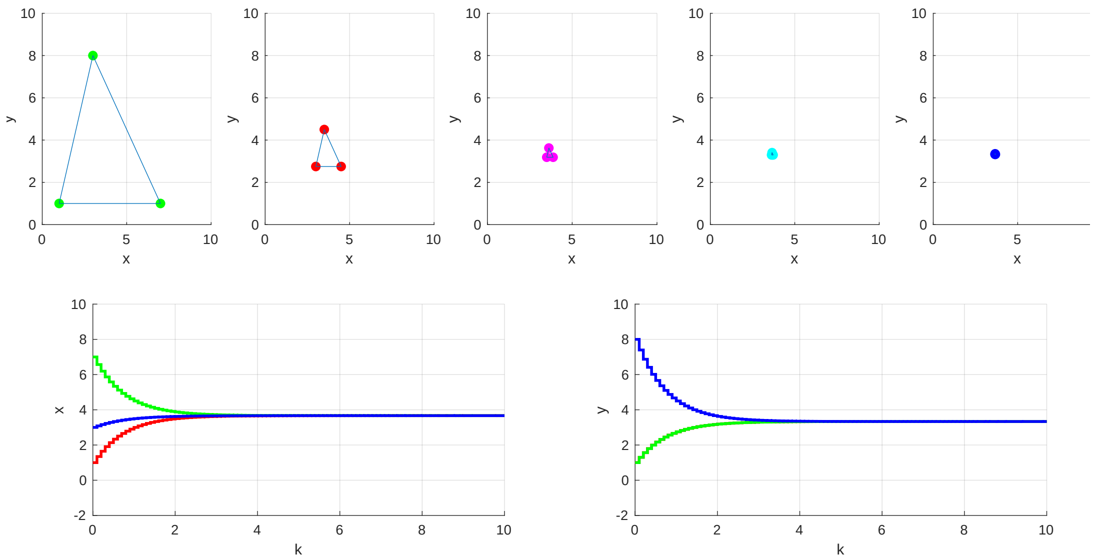
Case 1: $P = \begin{bmatrix} 0.50 & 0.50 & 0.00 \\ 0.00 & 0.50 & 0.50 \\ 0.50 & 0.00 & 0.50 \end{bmatrix}$, $\bar{P} = 0.33 \cdot \mathbb{1}\mathbb{1}^\top$, $\sigma(P) = \{1, 0.25 \pm 0.433\}$

Agents planar positions and shrinkage of the convex hull are as below (first iterations).



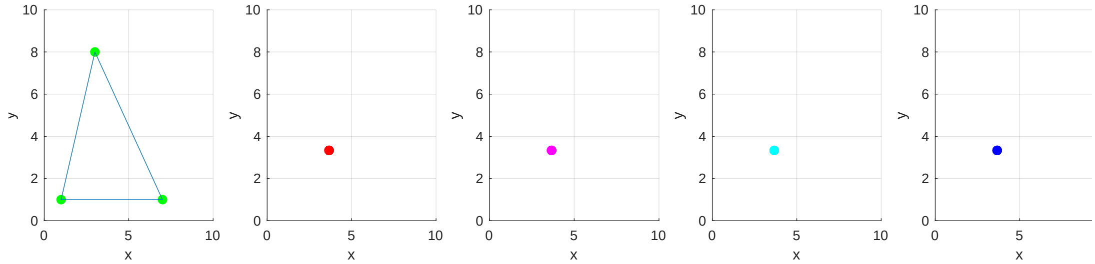
Case 2: $P = \begin{bmatrix} 0.50 & 0.25 & 0.25 \\ 0.25 & 0.50 & 0.25 \\ 0.25 & 0.25 & 0.50 \end{bmatrix}$, $\bar{P} = 0.33 \cdot \mathbb{1}\mathbb{1}^\top$, $\sigma(P) = \{1, 0.25, 0.25\}$

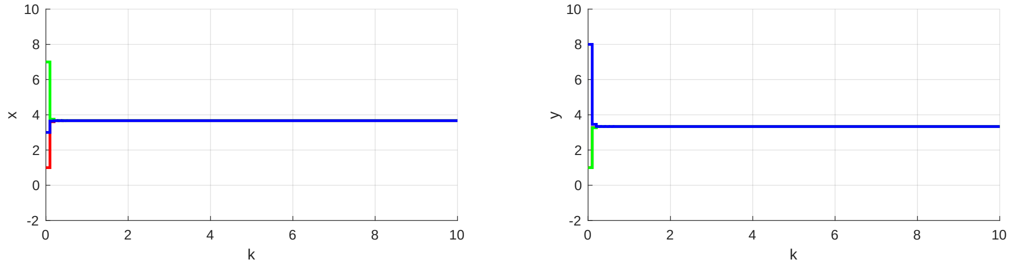
Agents planar positions and shrinkage of the convex hull are as below (first iterations).



Case 3: $P = \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}$, $\bar{P} = 0.33 \cdot \mathbb{1}\mathbb{1}^\top$, $\sigma(P) = \{1, 0, 0\}$

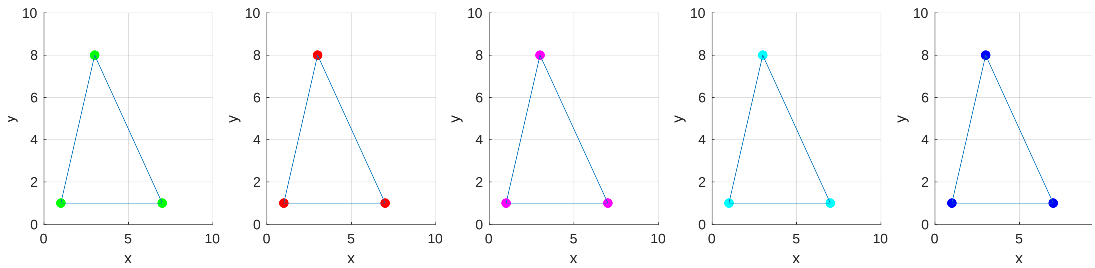
Agents planar positions and shrinkage of the convex hull are as below (first iterations).

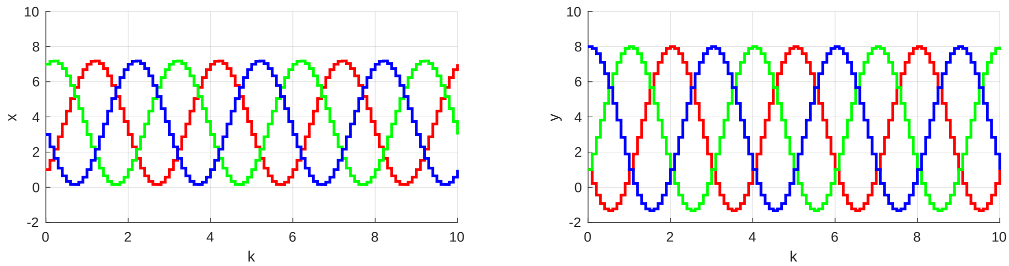




Case 4: $P = \begin{bmatrix} 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \\ 1.00 & 0.00 & 0.00 \end{bmatrix}$, $\bar{P} = N.A.$, $\sigma(P) = \{1, -0.5 \pm 0.866\}$

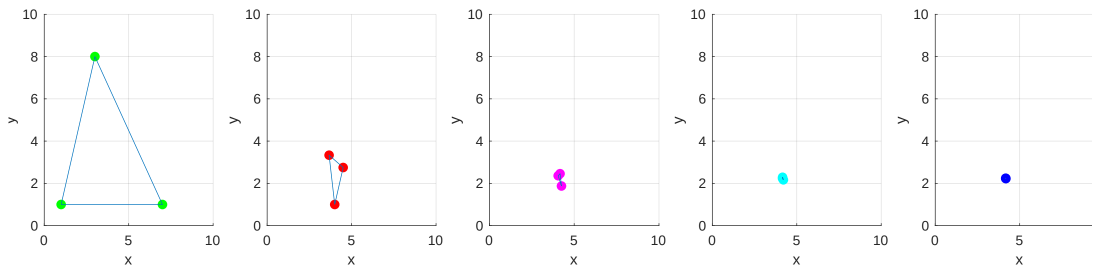
Agents planar positions and (no) shrinkage of the convex hull are as below (first iterations).

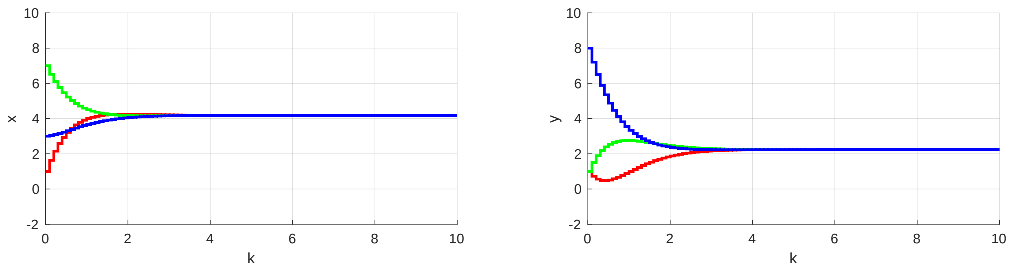




Case 5: $P = \begin{bmatrix} 0.50 & 0.50 & 0.00 \\ 0.25 & 0.50 & 0.25 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}$, $\bar{P} \neq 0.33 \cdot \mathbb{1}\mathbb{1}^\top$, $\sigma(P) = \{1, 0.1667 \pm 0.1179\}$

Agents planar positions and shrinkage of the convex hull are as below (first iterations).





The distributed estimation problem

On the other hand, for distributed estimation problems we could consider a network of n sensors measuring a common quantity, for example

$$x_i = x + w_i, \quad w_i \sim \mathcal{N}(0, \sigma^2)$$

where the centralized optimal value is given by

$$\hat{x} = \frac{1}{N} \sum_{i=1}^{N} x_i = \frac{\mathbb{1}^\top \mathbf{x}}{\mathbb{1}^\top \mathbb{1}}.$$

Even in this case, we consider the update law

$$\mathbf{x}(k+1) = P\mathbf{x}(k) \Rightarrow \mathbf{x}(k) = P^k \mathbf{x}(0) \tag{2.2}$$

If we suppose that $P^k \to \bar{P}$ for $k \to \infty$ we converge to the solution $\bar{P}\mathbf{x}(0)$.

1. to get $x_i \to \alpha$ for all $i = 1, \ldots, n$ (all sensors converge to the same value), we need to have $\bar{P}\mathbf{x}(0) = \alpha\mathbb{1}$, meaning that $\bar{P}$ has all equal rows. This is consensus.

2. to get $x_i \to \hat{x}$ for all $i = 1, \ldots, n$ (all sensors converge to the average value), we can note that by pre-multiplying on both sides $\bar{P}\mathbf{x}(0) = \hat{x}\mathbb{1}$ by $\mathbb{1}^\top$, we end up with $\mathbb{1}^\top \bar{P} = \mathbb{1}^\top$. So, even the columns of the $\bar{P}$ matrix sum to 1; then, the matrix is double stochastic. This is average consensus.

2.3 Basics on Graph Theory

The following are the main concepts of Graph Theory, a useful theoretical tool for studying Consensus Theory.

2.3.1 General definitions

Definition 11 (Graph) *An (undirected) graph is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \dots, v_n\}$ is the set of the nodes endowed with a state $\mathbf{x}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, i.e. connections between vertices.*

Definition 12 (Neighbours) *v_i, v_j are said to be neighbours iff $e_{ij} \in \mathcal{E}$.*

Definition 13 (Neighbourhood) *Given a vertex v_i, its neighbourhood is defined as the set $\mathcal{N}(v_i) = \mathcal{N}_i = \{v_j : e_{ij} \in \mathcal{E}\}$*

Definition 14 (Degree) *The number of neighbours of the node i is called is called degree of the node i.*

Definition 15 (Regular Graph) *A graph is said to be regular if all its nodes have the same degree.*

Definition 16 (Path) *A path is a sequence of neighbouring nodes. If it is closed it's called a* ***cycle****.*

Definition 17 (Connected Graph) *A graph is connected if $\forall v_i, v_j$ there exists a path connecting v_i with v_j.*

Definition 18 (Connective components) *The maximal subgraphs which are connected are called connected components.*

Definition 19 (Spanning Tree) *A spanning tree is a subgraph $\mathcal{G}$ that connects all the vertices in $\mathcal{V}$ with some of the edges in $\mathcal{E}$, without creating cycles.*

Definition 20 (Weighted Graph) *Let $w : \mathcal{E} \to \mathbb{R}$, a weighted graph is defined as the triplet $\mathcal{G}_w = (\mathcal{V}, w, \mathcal{E})$.*

This function allows to define the **length** of a path as $\sum_{i \in \text{path}} w_i$.

Definition 21 (Geodesic) *A geodesic is the path between two nodes with minimum length and the length of the longest geodesic is called* ***diameter****.*

Definition 22 (Oriented Graph) *An oriented graph (also called directed graph or digraph) is a graph where the edges have a starting and an ending node called tail and head, respectively.*

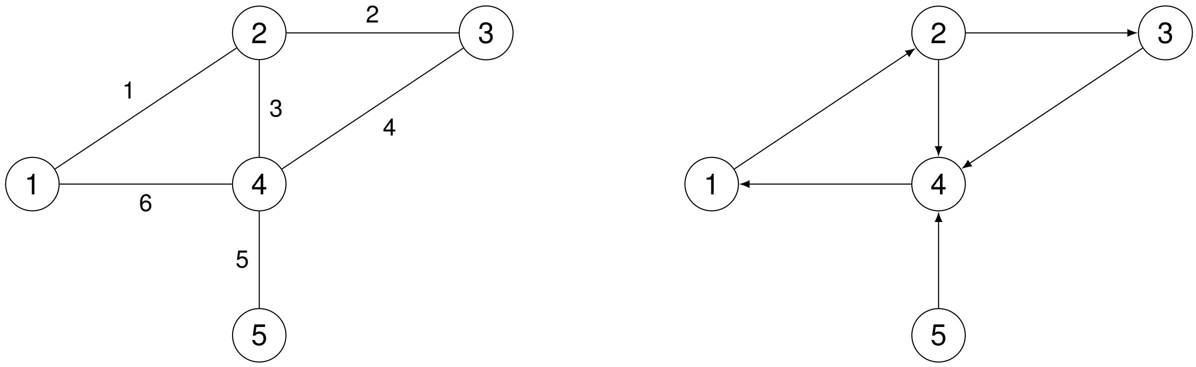
Definition 23 (Connected Digraph) *A digraph is said to be* ***strongly*** *connected if there exists an oriented path between any pair of nodes,* ***weakly*** *connected if the non-oriented version is connected.*

In the case of digraphs we distinguish between **in-degree**, i.e. the number of incoming edges of a node, and **out-degree**, i.e. the number of outgoing edges of a node. These distinction brings to define **in-neighbours** and **out-neighbours**.

Definition 24 (Self-Loops) *In order to model situations in which the nodes have a memory of their own state, we define self-loops, which are edges where tail and head coincide.*

2.3.2 Matrices defined over graphs

In this section, we define the matrices used to characterize a graph (directed and undirected) and their properties. The graph will be denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ number of nodes and $|\mathcal{E}| = m$ number of edges.



(a) Not-oriented graph (b) Oriented graph

Figure 2.1: Graph and digraph used as examples in the following

Definition 25 (Node Degree Matrix) *We define the node degree matrix of a (non-oriented) graph $\mathcal{G}$ as the diagonal matrix $\Delta_{\mathcal{G}} \in \mathbb{R}^{n\times n}$ with entries the degrees of the nodes. For oriented graphs, we define the diagonal matrices $\Delta_{\mathcal{G}_{IN}}$ and $\Delta_{\mathcal{G}_{OUT}}$, whose elements are, respectively, the in-degrees and out-degrees of the nodes.*

Consider, for example, the non-oriented and oriented graphs in figure 2.1. The node degree matrices for the graphs are:

$$\Delta_{\mathcal{G}} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ (non-oriented)}$$

$$\Delta_{\mathcal{G}_{\mathcal{OUT}}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \Delta_{\mathcal{G}_{\mathcal{IN}}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ (oriented).}$$

For undirected graphs, the following Lemma holds:

Lemma 1 (Handshaking Lemma) *Let $d_i,\ i = 1, \dots, n$ be the node degrees, then:*

$$\sum_{\mathcal{V}} d_i = 2 \cdot |\mathcal{E}|. \tag{2.3}$$

Lemma 1 also states that the number of nodes with odd degree is even (the overall number of "handshakes" between nodes with odd degree is always even).

Now, we define a matrix that represents the relations among nodes in the graph.

Definition 26 (Adjacency Matrix) *The adjacency matrix $A_{\mathcal{G}} \in \mathbb{R}^{n\times n}$ of an undirected graph without self-loops is defined as:*

$$A_{\mathcal{G}}(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}; \tag{2.4}$$

$$A_{\mathcal{G}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \textit{(non-oriented)}$$

whereas for a directed graph without self-loops it is:

$$A_{\mathcal{G}}(i,j) = \begin{cases} 1 & \text{if } (i \to j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}. \tag{2.5}$$

$$A_{\mathcal{G}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \textit{(oriented)}$$

Note that in the undirected case matrix $A_{\mathcal{G}}$ is symmetric and in the directed case it is, in general, non-symmetric.

Remark 10 *We have not considered the self-loops so far. In this case, if the graph has self-loops, we can extend the definition by choosing:*

$$A_{\mathcal{G}}(i,i) = 1 \text{ or } 2 \text{ if } (i,i)/(i \to i) \in \mathcal{E} \tag{2.6}$$

With the incidence matrix we state the relations between nodes and edges in the graph.

Definition 27 (Incidence Matrix) *We define the incidence matrix $D_{\mathcal{G}} \in \mathbb{R}^{n\times m}$ of an undirected graph as follows:*

$$D_{\mathcal{G}}(i,k) = \begin{cases} 1 & \text{if } v_i \in e_k \\ 0 & \text{otherwise} \end{cases}; \tag{2.7}$$

$$D_{\mathcal{G}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \textit{(non-oriented)}$$

and, for directed graphs:

$$D_{\mathcal{G}}(i,k) = \begin{cases} +1 & \text{if } v_i \text{ tail of } e_k \\ -1 & \text{if } v_i \text{ head of } e_k \\ 0 & \text{otherwise} \end{cases}. \tag{2.8}$$

$$D_{\mathcal{G}} = \begin{bmatrix} +1 & 0 & 0 & 0 & 0 & -1 \\ -1 & +1 & +1 & 0 & 0 & 0 \\ 0 & -1 & 0 & +1 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & +1 \\ 0 & 0 & 0 & 0 & +1 & 0 \end{bmatrix} \text{ (oriented)}$$

Note that the sum of the elements in a column is always: 2 in the undirected case, 0 in the directed case.

2.3.3 Properties of undirected graphs

Consider now only the case of undirected graphs without self-loops.

Definition 28 (Laplacian of a Graph) *We define the Laplacian of a graph $L_{\mathcal{G}} \in \mathbb{R}^{n\times n}$ as:*

$$L_{\mathcal{G}} = \Delta_{\mathcal{G}} - A_{\mathcal{G}} \ . \tag{2.9}$$

$$L_{\mathcal{G}} = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & -1 & -1 & 4 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

If we choose an arbitrary orientation of the edges, we can further characterize the Laplacian through its incidence matrix as follows:

$$L_{\mathcal{G}} = D_{\mathcal{G}} D_{\mathcal{G}}^{\top} \ . \tag{2.10}$$

Proposition 1 (Properties of the Laplacian Matrix) *Here, we define some notable properties of the Laplacian matrix.*

1. *If $L_{\mathcal{G}}$ is symmetric, then its eigenvalues λ are real.*
2. *The sum by rows and columns is equal to zero.*
3. *$L_{\mathcal{G}}\mathbb{1} = 0\mathbb{1}$, i.e. 0 is eigenvalue of the Laplacian with eigenvector $\mathbb{1}$.*
4. *$L_{\mathcal{G}}$ is positive semidefinite. Indeed, by the characterization of the Laplacian through the incidence matrix, and denoting the state of the graph with $\mathbf{x}$, it follows that:*

   $$\mathbf{x}^{\top} L_{\mathcal{G}} \mathbf{x} = \mathbf{x}^{\top} D_{\mathcal{G}} D_{\mathcal{G}}^{\top} \mathbf{x} = \left\| D_{\mathcal{G}}^{\top} \mathbf{x} \right\|_2^2 \geq 0 \ ;$$

   therefore the eigenvalues are all non-negative.
5. *By combining properties 3) and 4) it follows that:*

   $$0 = \lambda_0(\mathcal{G}) \leq \lambda_1(\mathcal{G}) \leq \ldots \leq \lambda_{n-1}(\mathcal{G}). \tag{2.11}$$

The spectrum of the Laplacian $\sigma(L_{\mathcal{G}})$ will be also denoted as $\sigma(\mathcal{G})$ to characterize the **eigenvalues of the graph.** We present, now, some results for the spectral analysis of the Laplacian's eigenvalues.

Theorem 5 (Gershgorin Disks) *Given any square matrix $A \in \mathbb{R}^{n\times n}$, it holds that:*

$$\sigma(A) \subseteq \bigcup_{i=1,\dots,N} \left\{ z \in \mathbb{C} : \middle| |z - a_{ii}| \leq \sum_{j\neq i} |a_{ij}| \right\}; \tag{2.12}$$

i.e. the the spectrum of A is included in the union of all circles centered in $(a_{ii}, 0)$ of radius $\sum_{j\neq i} |a_{ij}|$ (see Fig. 2.2).

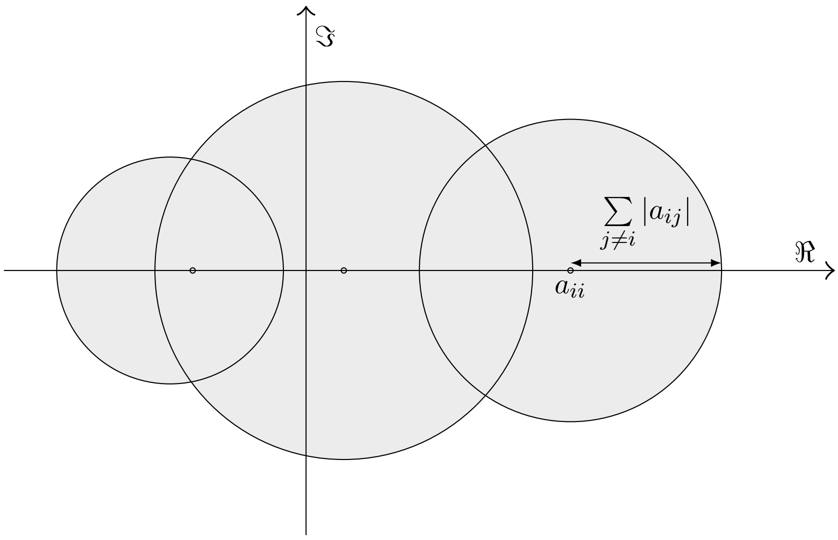


Figure 2.2: Gershgorin disks theorem: eigenvalues for matrix A lay in the grey area

Proof: Given an arbitrary λ and an arbitrary $x \neq 0$, $x = [x_1 \dots x_n]^\top$, choose i s.t. $|x_i| = \max_{j\in 1\dots n} |x_j| > 0$. We have that:

$$\lambda x_i = \sum_{j=1}^{n} a_{ij}x_j \Leftrightarrow \lambda x_i = a_{ii}x_i + \sum_{j\neq i} a_{ij}x_j,$$

which is true if and only if:

$$x_i(\lambda - a_{ii}) = \sum_{j\neq i} a_{ij}x_j \Leftrightarrow \lambda - a_{ii} = \sum_{j\neq i} a_{ij}\frac{x_j}{x_i}.$$

This implies the following chain of inequalities:

$$|\lambda - a_{ii}| = \left| \sum_{j\neq i} a_{ij}\frac{x_j}{x_i} \right| \leq \sum_{j\neq i} |a_{ij}|\frac{|x_j|}{|x_i|} \leq \sum_{j\neq i} |a_{ij}|$$

and in conclusion:

$$|\lambda - a_{ii}| \leq \sum_{j\neq i} |a_{ij}|.$$

This equation defines a set of possible locations for λ. Recalling that λ is arbitrary, taking the union of these sets we obtain the possible locations of any eigenvalue as showed in Fig. 2.2. □

Let's see an application of the Gershgorin Theorem to the Laplacian matrix. We know that the sum by rows for the Laplacian matrix is equal to zero. Therefore the circles defined in the Gershgorin Theorem are centered in $(d_i, 0)$, where d_i is the degree of the i-th node, and have radius equal to d_i. It follows that the eigenvalues of the Laplacian are all in the interval $[0, 2d_{MAX}]$, where $d_{MAX} = max\{d_i, i = 1, \ldots, N\}$; i.e. they satisfy the relation:

$$0 = \lambda_0 \leq \ldots \leq \lambda_{n-1} \leq 2d_{MAX}. \tag{2.13}$$

For the example in figure 2.1(a), we have $\sigma(L_{\mathcal{G}}) = \{0, 1, 2, 4, 5\}$ and $2d_{MAX} = 8$. The placement of eigenvalues is shown in figure 2.3. The red crosses indicate the eigenvalues.

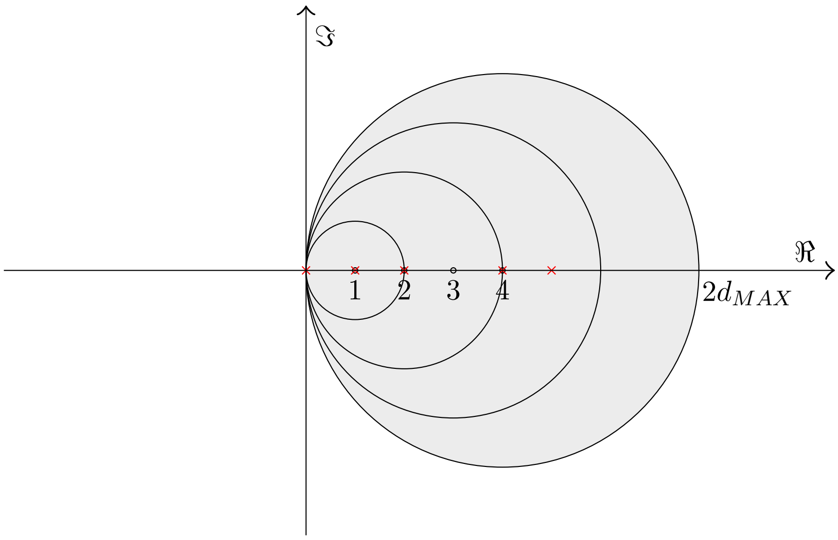


Figure 2.3: Eigenvalues of the Laplacian matrix lay in the grey area

Definition 29 (Potential of a Graph) *The potential of a graph is defined as:*

$$\varphi_{\mathcal{G}}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top L_{\mathcal{G}}\mathbf{x}. \tag{2.14}$$

For the potential the following result holds.

Theorem 6 (Potential of the Graph) *Given a non-oriented graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, it holds that:*

1. *The Laplacian and the potential are positive semidefinite.*
2. *It is given by:*
$$\mathbf{x}^\top L_{\mathcal{G}}\mathbf{x} = \sum_{\forall e_{ij} \in \mathcal{E}} (x_i - x_j)^2. \tag{2.15}$$
3. *The potential of a connected graph is equal to zero if and only if $x_i = x_j$ for all i, j.*

Proof:

1. $\mathbf{x}^\top L_{\mathcal{G}}\mathbf{x} = \mathbf{x}^\top \mathcal{D}_{\mathcal{G}}\mathcal{D}_{\mathcal{G}}{}^\top\mathbf{x} = ||\mathcal{D}_{\mathcal{G}}{}^\top\mathbf{x}||_2^2 \geq 0$
2. $[\mathcal{D}_{\mathcal{G}}{}^\top\mathbf{x}]_{ij} = (x_i - x_j)_{e_{ij}} \Rightarrow ||\mathcal{D}_{\mathcal{G}}{}^\top\mathbf{x}||_2^2 = \sum_{\forall e_{ij} \in \mathcal{E}} (x_i - x_j)^2$

3. By equation 2.15, if $x_i = x_j$ for all $e_{ij} \in \mathcal{E}$ for a connected graph, then $\varphi_{\mathcal{G}}(\mathbf{x}) = 0$. On the other hand we have that $\varphi_{\mathcal{G}}(\mathbf{x}) = 0$ implies $\sum\limits_{\forall e_{ij} \in \mathcal{E}} (x_i - x_j)^2 = 0$. Therefore, $x_i = x_j$ on every edge, which means that $x_i = x_j$ for all (v_i, v_j).

$\square$

With the following theorem we state a condition for the connectivity of a graph.

Theorem 7 (Connectivity of a Graph) *A non-oriented graph $\mathcal{G}$ is connected if and only if $\lambda_1(\mathcal{G}) > 0$, where λ_1 denotes the second lowest eigenvalue of the graph and it is called the* ***Fiedler's value****.*

Proof: $\lambda_1(\mathcal{G}) > 0 \Rightarrow \mathcal{G}$ is connected: by contrapositive
(By contrapositive) We want to prove that if $\mathcal{G}$ is not connected $\Rightarrow \lambda_1(\mathcal{G}) = 0$. If $\mathcal{G}$ is not connected, it is the union of (at least) two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, with a block structured Laplacian matrix (after node reordering)

$$L_{\mathcal{G}} = \begin{bmatrix} L_{\mathcal{G}_1} & 0 \\ 0 & L_{\mathcal{G}_2} \end{bmatrix}$$

We have (at least) two orthogonal eigenvectors related to the null eigenvalue: $\begin{bmatrix} \mathbb{1} \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ \mathbb{1} \end{bmatrix}$, hence $dim(ker(L_{\mathcal{G}})) \geq 2$ and $\lambda_1(\mathcal{G}) = 0$.

$\mathcal{G}$ is connected $\Rightarrow \lambda_1(\mathcal{G}) > 0$: by contradiction
(By contradiction) Suppose that $\mathcal{G}$ is connected and $\lambda_1(\mathcal{G}) = 0$. $L_{\mathcal{G}}$ is diagonalizable and there are two eigenvectors associated with the null eigenvalue: $\mathbb{1}$ and $\bar{x} \notin <\mathbb{1}>$.
From $L_{\mathcal{G}}\bar{x} = 0$ it follows that $\bar{x}^\top L_{\mathcal{G}}\bar{x} = 0 \Rightarrow \sum_{\forall e_{ij} \in \mathcal{E}} (x_i - x_j)^2 = 0 \Rightarrow x_i = x_j$ for all $e_{ij} \in \mathcal{E}$.
But this last statement implies that $\bar{x} \in \mathrm{span}\{\mathbb{1}\}$ and so we have a contradiction. $\square$

From the theorem above it follows that, if a graph is connected, then it has only one zero eigenvalue. Furthermore, it can be proven as a corollary that *the number of null eigenvalues is equal to the number of connected components* in the graph. In the case of a graph with more than one connected component, the Laplacian matrix can be put into a block-diagonal form, following a permutation of the nodes.

Remark 11 *It is possible to state some connectivity bounds of particular interest; in a connected graph*

$$\lambda_1(\mathcal{G}) \leq k_{\mathcal{V}}(\mathcal{G}) \leq k_{\mathcal{E}}(\mathcal{G}) \leq \min\{d_i\} \tag{2.16}$$

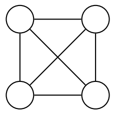
where
- $\lambda_1(\mathcal{G})$: is the first non-null eigenvalue;
- $k_{\mathcal{V}}(\mathcal{G})$: is the node connectivity*, namely the minimum number of nodes that can be removed to make the graph disconnected;*
- $k_{\mathcal{E}}(\mathcal{G})$: is the edge connectivity*, namely the minimum number of edges that can be cut to make the graph disconnected;*
- $\min\{d_i\}$: is the minimum node degree.

Theorem 8 (Kirchhoff's Theorem) *Given a connected graph $\mathcal{G}$ with n nodes, the number $ST_{\mathcal{G}}$ of spanning trees on $\mathcal{G}$ is given by:*

$$ST_{\mathcal{G}} = \frac{1}{n}\lambda_1 \cdot \lambda_2 \cdots \lambda_{n-1}. \tag{2.17}$$

We notice that, in order to have a large value of $ST_{\mathcal{G}}$, the $\{\lambda_i\}$ need to take larger values and, recalling that $0 < \lambda_i \leq 2d_{MAX},\ i = 1, \ldots, N$, this means that the nodes will have high degrees. Intuitively, the larger a node's degree is the more spanning trees are likely to include that node.

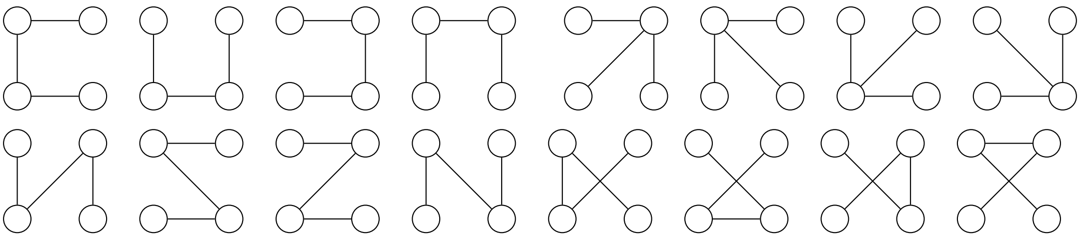
Example 10 *Consider the graph $\mathcal{G}$ with 4 nodes and 6 edges*



By computing the Laplacian matrix, we get

$$L_{\mathcal{G}} = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix} \qquad \sigma_{\mathcal{G}} = \{0, 4, 4, 4\} \qquad \Rightarrow \qquad ST_{\mathcal{G}} = 16$$

Indeed, we can find the following spanning trees



2.3.4 Properties of directed graphs

Given a digraph without self-loops, its Laplacian matrix is defined as:

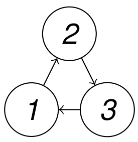
$$L_{\mathcal{G}} = \Delta_{\mathcal{G}_{OUT}} - A_{\mathcal{G}}. \tag{2.18}$$

In this case, though this matrix is not symmetric and positive semidefinite, the sum by rows is still equal to zero. Therefore 0 is an eigenvalue with corresponding eigenvector $\mathbb{1}$.
For the example in figure 2.1(b) we have

$$L_{\mathcal{G}} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 2 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

The Gershgorin theorem can be applied but, unlike the undirected case, the eigenvalues take in general complex values.
The Connectivity Theorem is modified as follows: if a digraph is *strongly connected* then rank$(L_{\mathcal{G}}) = n - 1$.

Example 11 *Consider the graph $\mathcal{G}$ with 3 nodes and 3 directed edges*

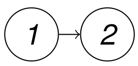


By computing the graph Laplacian matrix, we get

$$L_{\mathcal{G}} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \textit{rank}(L_{\mathcal{G}}) = n - 1 = 2$$

*and indeed the graph is strongly connected (*strongly connected $\Rightarrow$ rank condition*).*

Example 12 *Consider the graph $\mathcal{G}$ with 2 nodes and just 1 directed edge*



This graph is clearly not strongly connected but computing the Laplacian matrix, we still get

$$L_{\mathcal{G}} = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \qquad \textit{rank}(L_{\mathcal{G}}) = n - 1 = 1$$

*as a counterexample (*strongly connected $\not\Leftarrow$ rank condition*).*

2.4 Consensus matrix P and the Perron-Frobenius theory

2.4.1 Rationale

We consider the consensus protocol (in discrete time)

$$\mathbf{x}(k+1) = P\mathbf{x}(k)$$

and before stating the theorem that characterizes its convergence properties we provide some intuition behind that, by given some desiderata:

1. We want P such that there is a dominant eigenvalue λ_0 with a Jordan block of dimension 1. Note that in general P is non-negative.
2. Being $\sigma(P)$ the spectrum of P, $\rho(P)$ is the spectral radius of P ($\rho(P) = \max\{|\lambda| \text{s.t.} \lambda \in \sigma(P)\}$). We note that if $\begin{cases} \rho(P) \leq 1 \\ 1 \text{ is associated to block of dimension } 1 \end{cases}$ all modes related to the non-dominant eigenvalue vanish as the number of iterations increases.

Why do we ask for a dominant λ_0 with a Jordan block of dimension 1? In such a case, the Jordan decomposition results to be

$$P = T^{-1} \begin{bmatrix} \lambda_0 & 0 \\ 0 & \Lambda \end{bmatrix} T$$

and it follows that

$$P^k = T^{-1} \begin{bmatrix} \lambda_0^k & 0 \\ 0 & \Lambda^k \end{bmatrix} T = [v_0 \; V] \begin{bmatrix} \lambda_0^k & 0 \\ 0 & \Lambda^k \end{bmatrix} \begin{bmatrix} w_0^\top \\ W \end{bmatrix}$$

Note that

$$T^{-1}T = I_n \Rightarrow [v_0 \; V] \begin{bmatrix} w_0^\top \\ W \end{bmatrix} = \begin{bmatrix} w_0^\top \\ W \end{bmatrix} [v_0 \; V] = \begin{bmatrix} w_0^\top v_0 & w_0^\top V \\ W v_0 & WV \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & I_{n-1} \end{bmatrix}$$

Then

$$P^k = [v_0\lambda_0^k \quad V\Lambda^k] \begin{bmatrix} w_0^\top \\ W \end{bmatrix} = \underbrace{v_0\lambda_0^k w_0^\top}_{dominant} + \underbrace{V\Lambda^k W}_{dominated} \overset{k\to\infty}{\to} \bar{P} \approx v_0\lambda_0^k w_0^\top = \lambda_0^k (v_0 w_0^\top)$$

Example 13 *Consider a complete graph with self-loops:*

$$P = \begin{bmatrix} 1 & \cdots & 1 \\ 1 & \cdots & 1 \\ 1 & \cdots & 1 \end{bmatrix} = \mathbb{1}\mathbb{1}^\top$$

with

$\lambda_0 = n, \lambda_{1,\dots,n-1} = 0 : \lambda_0$ *is the dominant eigenvalue*

It follows

$$P^k = (\mathbb{1}\mathbb{1}^\top)^k = \underbrace{\mathbb{1}\mathbb{1}^\top \cdots \mathbb{1}\mathbb{1}^\top}_{k\ times} = \mathbb{1}\,\underbrace{\mathbb{1}^\top \cdots \mathbb{1}}_{k-1\ times}\,\mathbb{1}^\top = \mathbb{1}(\underbrace{\mathbb{1}^\top\mathbb{1}}_{n})^{k-1}\mathbb{1}^\top = n^k\frac{\mathbb{1}\mathbb{1}^\top}{n}$$

Note that the Jordan form P_j of P is

$$P_j = \left[\begin{array}{c|ccc} n & & & \\ \hline & 0 & & \\ & & \ddots & \\ & & & 0 \end{array}\right] \Rightarrow \begin{cases} \lambda_0 = n \\ v_0 = \mathbb{1} \\ w_0 = \dfrac{\mathbb{1}}{n} \end{cases}$$

With respect to the stochastic matrices that appear to be our choice for the consensus law, we have the following

Lemma 2 *If P is a stochastic matrix then*

1. *1 is an eigenvalue with the associated eigenvector $\mathbb{1}$*
2. *$\sigma(P)$ is a subset of the unit disk with $\rho(P) = 1$.*

Proof: Since P stochastic $P_{ij} \geq 0 \rightarrow$ non-negative and $P\mathbb{1} = \mathbb{1} \rightarrow (1, \mathbb{1})$ is an eigenpair.
From the application of Gershgorin Disks Theorem to P stochastic, we have that the disks are all passing through 1, centered in the points $(P_{ii}, 0)$ and with radius equal to the sum of the other terms row P_{ij}, $i \neq j$.
In other words, since P is stochastic, the domain for the eigenvalues is within the unit circle (included) (and note that P is not an asymptotically stable matrix). □

2.4.2 Perron-Frobenius theory

A first simple classification of non negative matrices is as follows:

Definition 30 *Given a generic matrix $P \in \mathbb{R}^{n\times n}$,*

- *if we have P such that $P_{ij} > 0$, then we say that P is positive: $P > 0$;*
- *if we have P such that $P_{ij} \geq 0$, then we say that P is non negative: $P \geq 0$.*

and more in detail:

Definition 31 *A generic non negative matrix $P \in \mathbb{R}^{n\times n}$ is:*

1. *Irreducible if and only if*
$$\sum_{k=0}^{n-1} P^k > 0 \tag{2.19}$$
2. *Primitive if and only if $\exists \bar{k} \in \mathbb{N}$ such that*
$$P^{\bar{k}} > 0$$

In particular, we will see that if P is primitive then it is also irreducible but not viceversa.

The important inclusion relation among these sets of matrices is shown in figure 2.4.

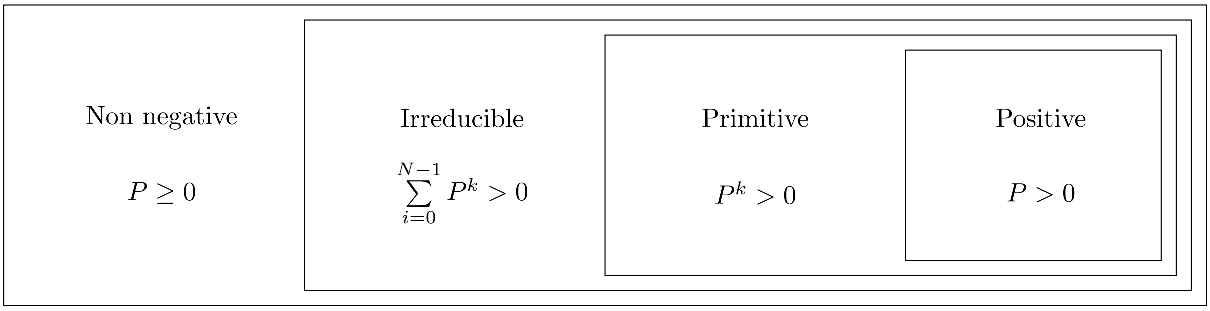


Figure 2.4: Relations among non-negative matrices.

Some examples are:

1. Non negative matrix:
$$P = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; \tag{2.20}$$

2. Irreducible matrix:
$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{in fact: } P^0 + P^1 > 0; \tag{2.21}$$

3. Primitive matrix:
$$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{in fact: } P^2 > 0; \tag{2.22}$$

4. Positive matrix:
$$P = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{2.23}$$

The fundamental theorem that provides a spectral characterization to these matrices is the following

Theorem 9 (Perron-Frobenius) *Consider a matrix $P \in \mathbb{R}^{n\times n}$.*

- *If P is non negative: there exists $\lambda_0 \geq 0$ such that*
  1. $\lambda_0 \geq |\lambda_i|, \quad \forall i = 1, ..., n-1$*;*
  2. *Right / left eigenvectors of P, (v_0, w_0) can be selected as non negative vectors, $v_0, w_0 \geq 0$.*
- *If P is irreducible:*
  1. *λ_0 is strictly positive, i.e. $\exists! \lambda_0 > 0$ such that $\lambda_0 \geq |\lambda_i|$;*
  2. *Right / left eigenvectors of P, (v_0, w_0) are unique and positive, $v_0, w_0 > 0$.*
- *If P is primitive:*
  1. *$\exists! \lambda_0 > 0$ such that $\lambda_0 > |\lambda_i|$.*
  2. *Right / left eigenvectors of P, (v_0, w_0) are unique and positive, $v_0, w_0 > 0$.*

Some observations about the theorem:

- λ_0 is the dominant eigenvalue and defines the spectral radius of matrix P;
- If the last property of the theorem holds, then it implies all the other properties; it states also that λ_0 is unique;

Example 14 *With respect to the matrices given before:*

- *Matrix (2.20) is non-negative and its spectrum is given by $\sigma = \{0, 0\}$. Here we have $\lambda_0 = 0$: it respects the condition 1 of the theorem, in fact it's greater or equal to the other eigenvalues but it's not unique.*

  Condition 2 states that we can select the eigenvectors: in this case they are all related to zero but we can select the one with all non-negative entries.

- *Matrix(2.21) is irreducible with spectrum $\sigma = \{1, -1\}$. In this case, as in codition 3, we have only one $\lambda_0 > 0$, but it is not strictly greater than the modulus of any other eigenvalues.*

  In this case we have that the eigenvectors are unique and their entries are positive.

- *Matrix(2.22) is primitive with spectrum given by $\sigma = \{1.61, -0.61\}$. As the last condition says, we have only one dominant eigenvalue $\lambda_0 = 1.61$ that is strictly larger than all the others.*

2.4.3 Application to the consensus problem

Consider the linear system

$$\mathbf{x}(k+1) = P\mathbf{x}(k). \tag{2.24}$$

Theorem 10 *If P is primitive, with λ_0 dominant eigenvalue and v_0, w_0 eigenvectors (normalized such that $w_0^\top v_0 = 1$), then*

$$\lim_{k\to+\infty} \frac{P^k}{\lambda_0^k} = v_0 w_0^\top. \tag{2.25}$$

In practice, this theorem has been proved previously when we stated λ_0 as dominant eigenvalue and we decomposed P in Jordan form ending up with a matrix $v_0\lambda_0^k w_0^\top$ as the convergence matrix. A direct consequence of this theorem is the following corollary:

Corollary 1 *If P is primitive and stochastic (by rows)*

1. *$\lambda_0 = \rho(P) = 1$ is simple, $\lambda_0 > |\lambda_i|$, then we have semi-convergence;*
2. *$\bar{P} = \lim_{k\to+\infty} P^k = \mathbb{1}w_0^\top$, where w_0 is normalized according to $\sum_{i=1}^{n} w_{0i} = 1$;*
3. *Given the update rule (2.24), we have $\lim_{k\to+\infty} P^k\mathbf{x}(0) = \alpha\mathbb{1}$, where $\alpha = w_0^\top\mathbf{x}(0)$ is a linear combination of the initial conditions with coefficients given by w_0 (**consensus**);*
4. *If P is doubly stochastic, then $w_0 = \frac{\mathbb{1}}{n}$ and $\lim_{k\to+\infty} \mathbf{x}(k) = \hat{\mathbf{x}}(0)\mathbb{1}$, where $\hat{\mathbf{x}}(0) = w_0^\top\mathbf{x}(0)$ is the average of the initial condition (**average consensus**).*

Remark 12 *The following considerations are in order*

- *Note that, in property 3, we have that $\mathbb{1}w_0^\top x(0) = (w_0^\top x(0))\mathbb{1}$. The value α is a scalar number and thus we can say that, when k tends to infinity, we converge to $\alpha\mathbb{1}$ which is the consensus. Thinking about the rendez-vous problem, this was the condition we had in all cases when we had the consensus, not only to the barycenter but to a generic point.*
- *We stress that from the point of view of iterative algorithms, $x(0)$ is the initial condition. Moreover, we think of (2.24) as the estimation of a common variable that is the measurement we take at the sampling instant. An interesting role is played by w_0: it is the weighting factor of the measurements we take.*

  In the case of social dynamics, w_{0i} is called the social influence of agent i.

Remark 13 *Some additional notes about the corollary:*

1. *The corollary gives us sufficient conditions.*

   Example 15 *Consider a reducible matrix*

$$\begin{cases} x_1(k+1) = x_1(k) \\ x_2(k+1) = x_2(k) \end{cases} \quad \Rightarrow \quad P = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}. \tag{2.26}$$

   The state converges to $x_1(0)\mathbb{1}$ with both variables.
   The matrix is not primitive, irreducible or positive but we still have convergence to a value that is a linear combination of the two states: $\mathbb{1}x_1 + 0x_2$.

2. *Positive matrixes (particular case of primitives) in graph theory imply self-loops. In fact, if P is positive, the diagonal is strictly positive. When we take the update rule for the i-th node, we surely have some self contribution: it means the system has memory.*

We complete by remarking some simple proofs of the statements in the corollary.
Consider P primitive and row stochastic. Consider the update rule (2.24). As we did before, we can decompose it in Jordan form as

$$P = T^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \Lambda \end{bmatrix} T = [v_1 \ V] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} w_1^\top \\ W \end{bmatrix} \tag{2.27}$$

where $v_1 = \mathbb{1}$ and $\lambda_1 = 1$.
Proof of corollary statement 2:
It is:

$$P^k = v_0 \lambda_0 w_0^\top + V \Lambda^k W. \tag{2.28}$$

Since all eigenvalues are such that $|\lambda_i| < 1$ for $i > 1$, the second part vanishes as k tends to infinity, thus $P^k \to \mathbb{1} w_0^\top$ when $k \to +\infty$.
Proof of corollary statement 3:

Given (2.24), as k increases we have $\bar{\mathbf{x}} = (\mathbb{1} w_0^\top)\mathbf{x}(0) = \begin{bmatrix} \alpha \\ \vdots \\ \alpha \end{bmatrix}$

Proof of corollary statement 4:

If $w_0 = \frac{\mathbb{1}}{n}$, then $\bar{\mathbf{x}} = \frac{(\mathbb{1} * \mathbb{1}^\top)\mathbf{x}(0)}{n} = \begin{bmatrix} \hat{x} \\ \vdots \\ \hat{x} \end{bmatrix}$ with $\hat{x} = \dfrac{1}{n} \sum_{i=1}^{n} x_i(0)$

2.4.4 Convergence rate of consensus

How fast is the state converging to consensus? We have that

$$P^k = \mathbb{1} w_0^\top + V \Lambda^k W \quad \longrightarrow \quad \bar{P} = \mathbb{1} w_0^\top. \tag{2.29}$$

We can define a convergence gap as

$$e^2(k) = ||x(k) - \bar{x}||_2^2 = ||(\mathbb{1} w_0^\top + V\Lambda^k W)\mathbf{x}(0) - (\mathbb{1} w_0^\top)\mathbf{x}(0)||_2^2 = ||(V\Lambda^k W)\mathbf{x}(0)||_2^2. \tag{2.30}$$

This convergence is ruled by Λ^k: it depends on the spectrum of Λ and in particular on the slowest of its modes (second largest eigenvalue of P). The situation is depicted in Fig. 2.5.

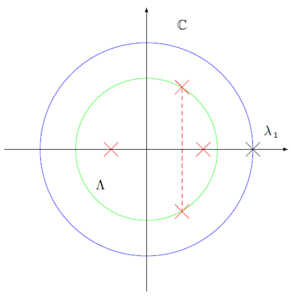


Figure 2.5: Spectral radius and spectrum of Λ.

2.4.5 Graphs and consensus

In this section we give some detail on how the graph theory and the consensus theory are strongly related; in particular, we'll consider a multiagent system ruled by the consensus protocol characterized by P and whose agents are connected through a network characterized by an adjacency matrix A.
What does it mean to have P primitive? Why do we need it?
Assume that the consensus matrix P is positive. Then, the adjacency matrix A is positive and it has the same support of the matrix P where the support of a matrix is defined as the set of the couples of indices corresponding to entries that are non-zero. More in general, we know that P positive is a particular case of P primitive. We can see that:

1. $P > 0 \quad \Rightarrow \quad A > 0$, where A is the adjacency matrix and the two matrixes have the same support. $P > 0$ means that P is full, that is any node in the graph is connected to another node.
   A gives us the number of paths (sequences of edges) of length 1;
2. $P^k > 0 \quad \Rightarrow \quad A^k > 0$. This matrix gives the number of paths of length k among nodes.

Example 16 *Consider the graph in figure 2.6, in which $m = |\mathcal{E}| = 6$ and $n = |\mathcal{V}| = 5$*

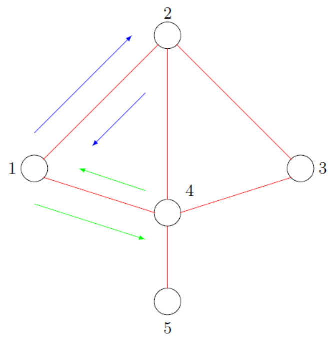


Figure 2.6: Graph

Consider the adjacency matrix A

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \longrightarrow A^2 = \begin{bmatrix} 2 & 1 & 2 & 1 & 1 \\ 1 & 3 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 & 1 \\ 1 & 2 & 1 & 4 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \ldots \longrightarrow A^4 > 0 \tag{2.31}$$

Note that, in matrix A^2, the entry (i, j) gives the number of paths of length 2 in the graph $\mathcal{G}$, from the node i to the node j. For example $A(1, 1) = 2 \Rightarrow$ there are 2 different paths that go from node 1 to node 1 in two steps (blue and green arrows in figure 2.6); $A(4, 2) = 2 \Rightarrow$ there are 2 different paths that go from node 4 to node 2 in two steps.
By iterating A^3 has as entry (i, j) the number of paths of length 3 that go from node i to node j and in general A^k has as entry (i, j) the number of paths of length k that go from node i to node j.

If we compute the successive powers of the matrix A we find that the first value of k for which A^k is full (i.e. has all its entries different from zero) is $k = 4$. This result implies that in 4 steps we can reach any other node from an arbitrary node of the graph $\mathcal{G}$. Note that, since $A^4 > 0$, then, by definition, the matrix $A_{\mathcal{G}}$ is primitive.
If the matrix is full, the graph will spread information on all the nodes, thus we will be able to reach every node in the graph starting from any node.
This concept can be also applied to the P matrix. So, we can state that if P is primitive then the power of A, sooner or later, will be full.

Example 17 *Consider now a digraph*

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \longrightarrow \quad A^2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} ... \tag{2.32}$$

At step 2 we have only 7 non-zero entries. Moreover, note that the last column remains and will remain null: this means that the information in node 5 will not be shared with the others. $A^1 0$ in non-null with the first 4 columns (last is null).
Consider now this slight modification, adding one self-loop (node 1) to the graph

$$A = \begin{bmatrix} \mathbf{1} & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \longrightarrow \quad A^2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} ... \tag{2.33}$$

We can see that just one self-loop has lead to 10 non-null entries: in particular, the self-loop in a connected graph provides a "parking place" where to stop along the path and this allows to equalize the path length more easily. A^4 in non-null with the first 4 columns (last is null: the self-loop is not solving the issue related to node 5).

2.4.6 More on positive, primitive, irreducible matrices

We have seen that

- Primitive matrix: $A^{\bar{k}} > 0 \Rightarrow \left[A^{\bar{k}}\right]_{ij} > 0$: for the same value of the exponent all entries will be positive
- Irreducible matrix: $\sum_0^{n-1} A^k > 0 \Rightarrow \left[A^k\right]_{ij} > 0$: for some value of the exponent (between 0 and $n-1$) entries will be positive, but not all together for the same $\bar{k}$.

We have the following definition:

Definition 32 (Primitivity exponent) *Given a primitive matrix $A \in \mathbb{R}^{n \times n}$, the minimum value $\bar{k}$ such that $A^{\bar{k}} > 0$ is called primitivity exponent:*

$$\bar{k} = \min_{k \in \mathbb{N} \setminus \{0\}} \{k \; s.t. \; A^k > 0\} \tag{2.34}$$

Remark 14 *In general, the primitivity exponent may be larger than the dimension n of the matrix. A good bound for the primitivity exponent is given as $\bar{k} \leq (n-1)^2 + 1$, that is exactly met for particularly sparse graphs.*

Example 18 *An example of a matrix A whose primitivity exponent satisfies the above bound with equality is:*

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

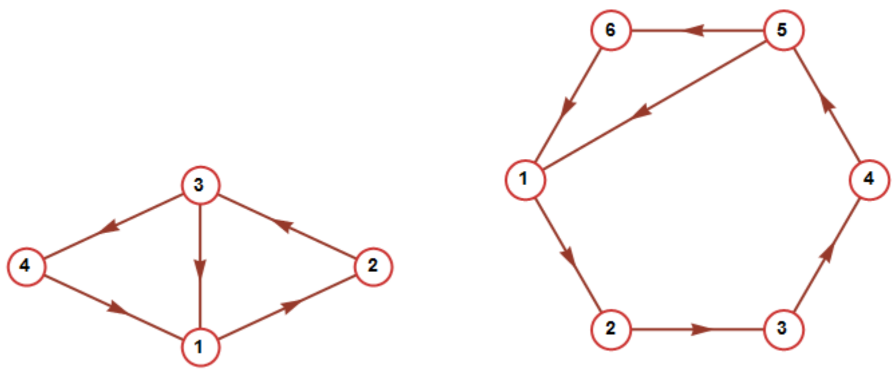
which is associated to the graph of Fig. 2.7(a). In fact, the first positive power of this matrix is the tenth one which is

$$A^{10} = \begin{bmatrix} 3 & 1 & 1 & 2 \\ 3 & 3 & 1 & 1 \\ 2 & 3 & 3 & 1 \\ 1 & 1 & 2 & 1 \end{bmatrix}$$

and we have that $\bar{k} = (4-1)^2 + 1 = 10$. The previous example can be generalized by the $n \times n$ matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \ddots & 0 \\ 1 & 0 & \dots & 0 & \ddots & 1 \\ 1 & 0 & \dots & \dots & \dots & 0 \end{bmatrix}$$

which corresponds in the case $n = 6$ to a graph with the structure of Fig. 2.7(b). In particular, in this case $\bar{k} = (6-1)^2 + 1 = 26$.



(a) (b)

Figure 2.7: Graph corresponding to a primitive matrix P maximizing the primitivity exponent.

We also have the following properties:

- Primitive A: any node is connected to any other by a $\bar{k}$-length path (connected / strongly connected).
- Irreducible A: any node is connected to another by some k-length path (connected / strongly connected).
- If starting from an irreducible A we can find a place where to stand and wait, we can equalize all paths.

that we can formalize in the following theorems

Theorem 11 *$\mathcal{G}$ is strongly connected $\Leftrightarrow$ A is irreducible*

Theorem 12 *A is irreducible and there is a self-loop in the graph $\mathcal{G}$ described by $A \Rightarrow A$ is primitive*

Remark 15 *If we have a matrix that describes a strongly connected graph with a self loop (an element of the diagonal is different from zero), the presence of a self loop allows us to make some paths longer, so it is possible to equalize all the paths' lengths and get the primitivity property. Conversely, note that a primitive matrix may not have self loops.*

In this sense, see the following examples:

Example 19 *Let's consider the graph of Fig. 2.8*

Figure 2.8: Graph without self loops.

We compute the adjacency matrix, its first 2 powers

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad A^2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad A^3 = \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 2 \\ 0 & 2 & 0 \end{bmatrix}$$

and the spectrum of A

$$\sigma(A) = \{+\sqrt{2}; -\sqrt{2}; 0\}$$

that lead to the following results:

$$\sum_{k=0}^{N-1} A^k > 0 \quad \Rightarrow \quad A \textit{ irreducible}$$

$$A^k \not> 0 \quad \Rightarrow \quad A \textit{ not primitive}$$

If we take the same graph of before, but with a self loop on node 1 (as in Fig. 2.9), we have:

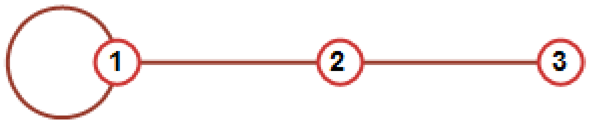


Figure 2.9: Graph with self loop.

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad A^2 = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad A^3 = \begin{bmatrix} 3 & 3 & 1 \\ 3 & 1 & 2 \\ 1 & 2 & 0 \end{bmatrix} \quad A^4 = \begin{bmatrix} 6 & 4 & 3 \\ 4 & 5 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

Here, A^4 is a positive matrix, $\bar{k}$ is larger than n and we don't reach the upper bound $(n-1)^2+1$ on the the primitivity exponent. We see that it is sufficient (not necessary) to have a self loop in the connected graph to have a graph corresponding to a primitive matrix.

Example 20 *Consider the matrix corresponding to the graph without self loops of Fig. 2.10*

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

If we compute A^2 we find

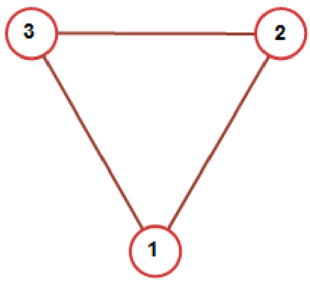


Figure 2.10: graph $\mathcal{G}$ corresponding to a matrix A with null diagonal elements.

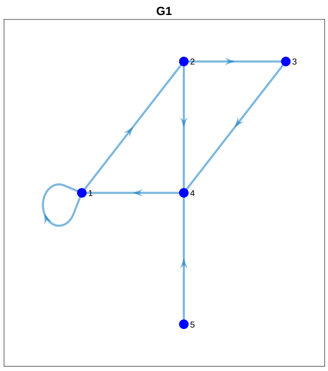
$$A^2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

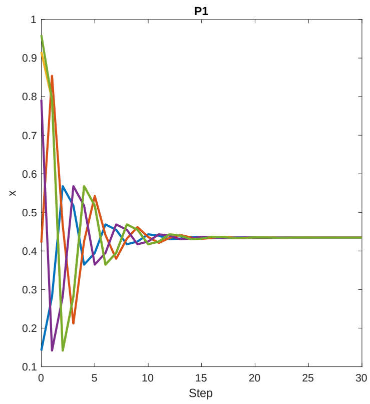
and therefore we have that the matrix A is primitive. This example shows that a primitive matrix A may have all its diagonal elements null and hence it can be associated to a graph without self loops.

Example 21 *Note that we can have consensus even if we do not have primitivity or irreducibility of the matrix governing the update rule. See the following:*

Case 1 $P = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ *- P is not irreducible - $P \to \bar{P} =$* $\begin{bmatrix} 0.4 & 0.2 & 0.1 & 0.2 & 0 \\ 0.4 & 0.2 & 0.1 & 0.2 & 0 \\ 0.4 & 0.2 & 0.1 & 0.2 & 0 \\ 0.4 & 0.2 & 0.1 & 0.2 & 0 \\ 0.4 & 0.2 & 0.1 & 0.2 & 0 \end{bmatrix}$

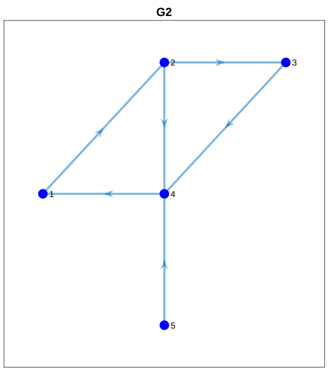
$\sigma_P = \{1, 0, -0.5, \pm 0.7i\}$ - $|\lambda_1| = 0.7$ *gives the convergence rate.*
We have consensus to some value (we do not consider state x_5).
Graph and convergence are reported below.

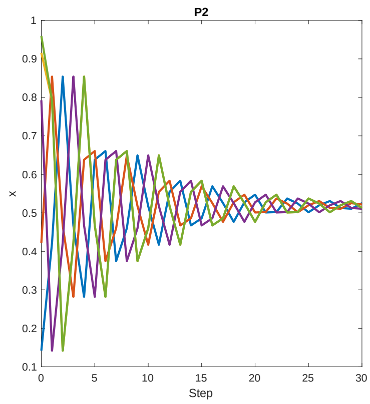




Case 2 $P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ *- P is not irreducible - $P \to \bar{P} =$* $\begin{bmatrix} 0.28 & 0.28 & 0.14 & 0.28 & 0 \\ 0.28 & 0.28 & 0.14 & 0.28 & 0 \\ 0.28 & 0.28 & 0.14 & 0.28 & 0 \\ 0.28 & 0.28 & 0.14 & 0.28 & 0 \\ 0.28 & 0.28 & 0.14 & 0.28 & 0 \end{bmatrix}$

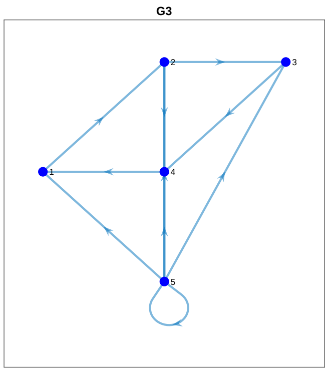
$\sigma_P = \{1, 0, -0, 64, -0.17 \pm 0.86i\}$ - $|\lambda_1| = 0.87$ *gives the convergence rate.*
We have consensus to some value (we do not consider state x_5).
We have slowed down consensus with respect to **Case 1**.
Graph and convergence are reported below.

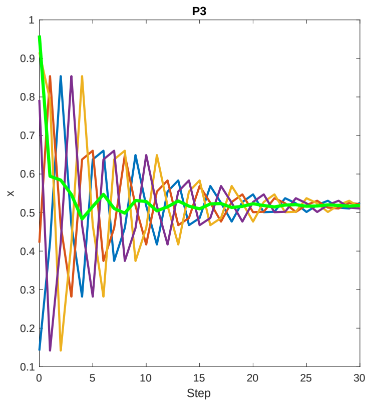




Case 3 $P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/8 & 1/8 & 1/4 \end{bmatrix}$ *- P is not irreducible -* $P \to \bar{P} = \begin{bmatrix} 0.28 & 0.28 & 0.14 & 0.28 & 0 \\ 0.28 & 0.28 & 0.14 & 0.28 & 0 \\ 0.28 & 0.28 & 0.14 & 0.28 & 0 \\ 0.28 & 0.28 & 0.14 & 0.28 & 0 \\ 0.28 & 0.28 & 0.14 & 0.28 & 0 \end{bmatrix}$

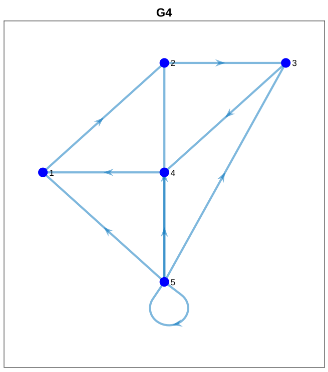
$\sigma_P = \{1, 0.25, -0, 64, -0.17 \pm 0.86i\}$ - $|\lambda_1| = 0.87$ *gives the convergence rate.*
We have consensus to some value (we do not consider state x_5).
We have speeded up consensus with respect to **Case 2** *only for node 5.*
Graph and convergence are reported below.

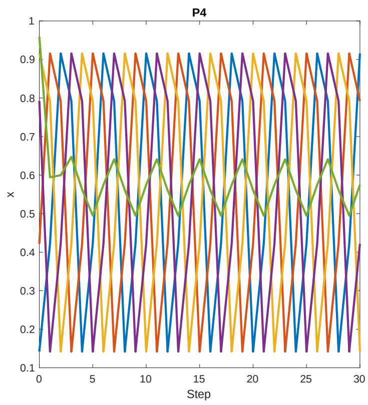




Case 4 $P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/8 & 1/8 & 1/4 \end{bmatrix}$ *- P is not irreducible -* $P \to \bar{P} = [?]$

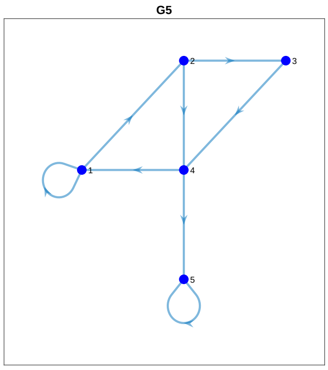
$\sigma_P = \{1, 0.25, -1, \pm i\}$ *- oscillatory behaviors.*
We do not have consensus.
Graph and no-convergence are reported below.

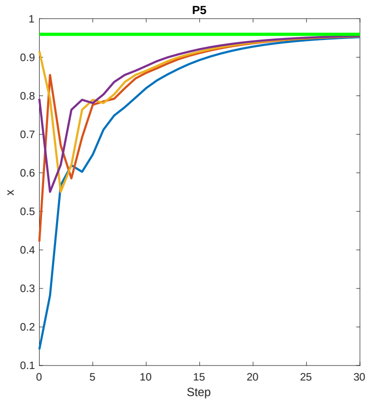




Case 5 $P = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ *- P is not irreducible -* $P \to \bar{P} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

$\sigma_P = \{1, -0.42, 0.86, 0.03 \pm 0.58i\}$ - $|\lambda_1| = 0.86$ *gives the convergence rate.*
We have consensus to the state x_5 imposed by node 5.
Graph and convergence are reported below.





More in general, we can state the following theorem that characterizes primitive matrices

Theorem 13 *If A is irreducible and the Greatest Common Divisor (GCD) of all cycles' lengths in the graph $\mathcal{G}$ described by A is equal to $1 \Leftrightarrow A$ is primitive*
The GCD of all cycles' lengths is called ***imprimitivity index****.*

Let's see some examples:

Example 22 *Trivially, if we have a graph with a self loop, $GCD = 1$ and the adjacency matrix is primitive.*

Example 23 *Consider the matrices below and the related graphs of figure 2.11:*

$$A_A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad A_B = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(a): the lengths of the cycles are 3 and 4 $\Rightarrow$ $GCD = 1$ and the matrix A_a is irreducible and primitive.
(b): the lengths of the cycles are 4 and 4 $\Rightarrow$ $GCD = 4$ and the matrix A_b is irreducible, but not primitive.

We may understand that, by adding edges, we increase the connectivity and the spreading of the information: the solution might converge faster, provided that the added edge does not cause loss of primitivity.

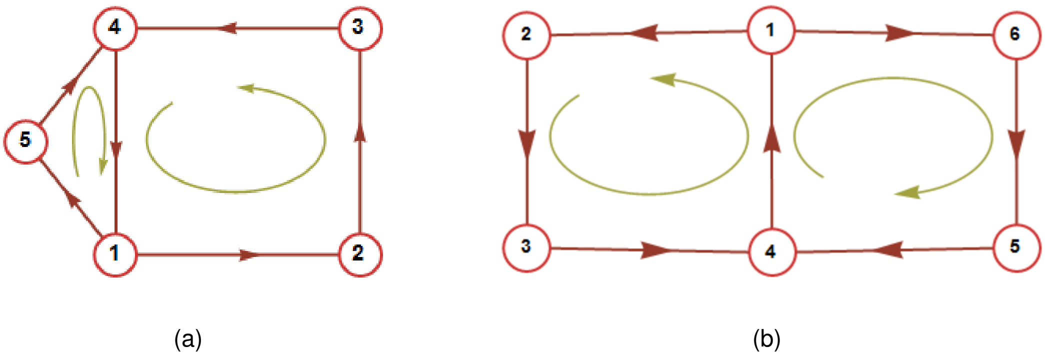


Figure 2.11: Graphs with: (a) $GCD = 1$ and (b) $GCD = 4$.

2.5 Design of the consensus matrix

In general, given a consensus problem we want to design P in order to:

1. P is adapted to the graph = $(\mathcal{V}, \mathcal{E})$ (same support as A);
2. the eigenvalues of P (except $\lambda_0 = 1$) are located as close as possible to 0. By doing so, we are locating also the second eigenvalue λ_1 that controls the speed of convergence.

Such properties are usually translated into a non convex optimization problem with the additional constraints of matrix stocasticity.
Therefore, we provide some heuristics that allow to provide a design guide for the P-matrix.

2.5.1 Laplacian-based design

Given the Laplacian associated to a generic graph $\mathcal{G}$

$$L_{\mathcal{G}} = \Delta_{\mathcal{G}} - A_{\mathcal{G}} \tag{2.35}$$

where $\Delta_{\mathcal{G}}$ is the degree matrix and $A_{\mathcal{G}}$ is the adjacency one. Remembering that the Laplacian $L_{\mathcal{G}}$ has a null eigenvalue, we consider

$$\begin{aligned} L_{\mathcal{G}}\mathbb{1} &= 0\mathbb{1} \\ I\mathbb{1} &= 1\mathbb{1} \end{aligned} \tag{2.36}$$

from which we obtain, introducing the positive parameter ϵ

$$(I - \epsilon L_{\mathcal{G}})\mathbb{1} = (1 - \epsilon 0)\mathbb{1} = \mathbb{1}, \qquad \epsilon > 0 \tag{2.37}$$

Let's define:

$$P \triangleq I - \epsilon L_{\mathcal{G}} \tag{2.38}$$

by looking at equation (2.37) we recognize the $\mathbb{1}$ right eigenvector and we need the condition on the entries of P to get a stochastic matrix:

$$\begin{cases} P_{ij} \geq 0 \\ \sum_j P_{ij} = 1 \end{cases} \tag{2.39}$$

By resorting to the definition of the Laplacian

$$P = \underbrace{(I - \epsilon\Delta_{\mathcal{G}})}_{\textit{diagonal entries}} + \underbrace{\epsilon A_{\mathcal{G}}}_{\textit{off diagonal entries}}, \tag{2.40}$$

where the off diagonal entries are always non negative (if ϵ is non-negative) and the diagonal entries are non-negative if and only if

$$1 - \epsilon d_i \geq 0 \Rightarrow 0 < \epsilon \leq \frac{1}{d_{max}}, \tag{2.41}$$

where d_{max} is the maximum degree of the nodes.

Observation 2 *This is a global method: indeed, we may not know the whole graph topology, and thus, we cannot derive the associated Laplacian in order to apply the previous algorithm.*