

# InMOOC

<https://www.ensta-bretagne.fr/jaulin/inmooc.html>

Luc Jaulin

August 29, 2020



# Contents

<b>1 Three-dimensional modeling</b>	<b>5</b>
1.1 Rotation matrices . . . . .	5
1.1.1 Definition . . . . .	6
1.1.2 Lie group . . . . .	7
1.1.3 Lie algebra . . . . .	7
1.1.4 Rotation vector . . . . .	8
1.1.5 Rodrigues rotation formulas . . . . .	10
1.1.6 Coordinate system change . . . . .	11
1.2 Euler angles . . . . .	13
1.2.1 Definition . . . . .	13
1.2.2 Rotation vector of a moving Euler matrix . . . . .	15
1.3 Inertial unit . . . . .	16
1.4 Dynamic modeling . . . . .	19
1.4.1 Principle . . . . .	19
1.4.2 Modeling a quadrotor . . . . .	20

A *mobile robot* can be defined as a mechanical system capable of moving in its environment in an autonomous manner. For that purpose, it must be equipped with:

- *sensors* that will collect knowledge of its surroundings (which it is more or less aware of) and determine its location ;
- *actuators* which will allow it to move ;
- an *intelligence* (or algorithm, regulator), which will allow it to compute, based on the data gathered by the sensors, the commands to send to the actuators in order to perform a given task.

Finally, to this we must add the *surroundings* of the robot which correspond to the world in which it evolves and its *mission* which is the task it has to accomplish. Mobile robots are constantly evolving, mainly from the beginning of the 2000s, in military domains (airborne drones [BEA 12], underwater robots [CRE 14], etc.), and even in medical and agricultural fields. They are in particularly high demand for performing tasks considered to be painful or dangerous to humans. This is the case for instance in mine-clearing operations, the search for black boxes of damaged aircraft on the ocean bed and planetary exploration. Artificial satellites, launchers (such as Ariane V), driverless subways and elevators are examples of mobile robots. Airliners, trains and cars evolve in a continuous fashion towards more and more autonomous systems and will very probably become mobile robots in the following decades.

*Mobile robotics* is the discipline which looks at the design of mobile robots [LAU 01]. It is based on other disciplines such as automatic control, signal processing, mechanics, computing and electronics. The aim of this book is to give an overview of the tools and methods of robotics which will aid in the design of mobile robots. The robots will be modeled by *state equations*, *i.e.*, first order (mostly non-linear) differential equations. These state equations can be obtained by using the laws of mechanics. It is not in our objectives to teach, in detail, the methods of robot modeling (refer to [JAU 05] and [JAU 13] for more information on the subject), merely to recall its principles.

# Chapter 1

## Three-dimensional modeling

This lesson looks at the three-dimensional modeling of a solid (non-articulated) robot. Such modeling is used to represent an airplane, a quadcopter, a submarine, and so forth. Through this modeling we will introduce a number of fundamental concepts in robotics such as state representation, rotation matrices and Euler angles. The robots, whether mobile, manipulator or articulated, can generally be put into a state representation form:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$

where  $\mathbf{x}$  is the state vector,  $\mathbf{u}$  the input vector and  $\mathbf{y}$  the vector of measurements [JAU 05]. We will call *modeling* the step which consists of finding a more or less accurate state representation of the robot. In general, constant parameters may appear in the state equations (such as the mass and the moment of inertia of a body, viscosity, etc.). In such cases, an identification step might prove to be necessary. We will assume that all of the parameters are known. Of course, there is no systematic methodology that can be applied for modeling a mobile robot. The aim of this chapter is to present the tools which allow to reach a state representation of three-dimensional solid robots in order for the reader to acquire a certain experience which will be helpful when modeling his/her own robots. This modeling will also allow us to recall a number of important concepts in Euclidean geometry, which are fundamental in mobile robotics. This chapter begins by recalling a number of important concepts in kinematics which will be useful for the modeling.

### 1.1 Rotation matrices

For three-dimensional modeling, it is essential to have a good understanding of the concepts related to rotation matrices, which are recalled in this section. It is by using this tool that we will perform our coordinate system transformations and position our objects in space.

### 1.1.1 Definition

Let us recall that the  $j^{\text{th}}$  column of the matrix of a linear application of  $\mathbb{R}^n \rightarrow \mathbb{R}^n$  represents the image of the  $j^{\text{th}}$  vector  $e_j$  of the standard basis (see Figure 1.1). Thus, the expression of a rotation matrix of angle  $\theta$  in the plane  $\mathbb{R}^2$  is given by:

$$\mathbf{R} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

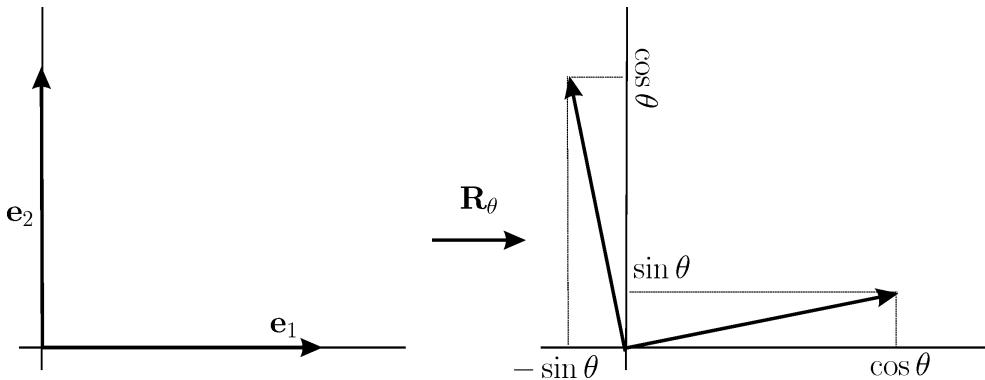


Figure 1.1: Rotation of angle  $\theta$  in a plane

Concerning rotations in the space  $\mathbb{R}^3$  (see Figure 1.2), it is important to specify the axis of rotation. We distinguish three main rotations: the rotation around the  $Ox$  axis, the one around the  $Oy$  axis and the one around the  $Oz$  axis.

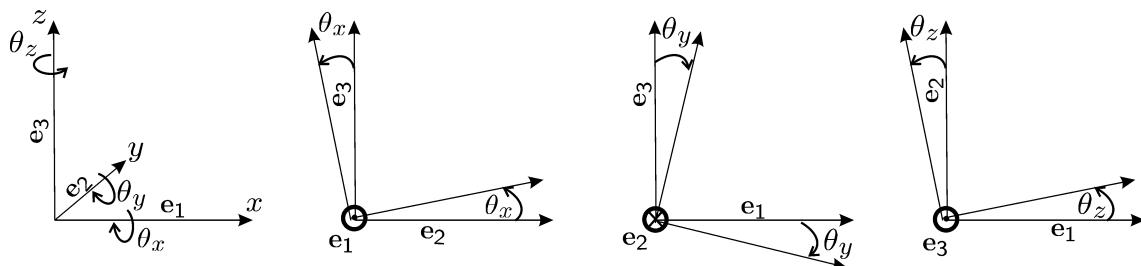


Figure 1.2: Rotations in  $\mathbb{R}^3$  following various viewing angles

The associated matrices are respectively given by:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{pmatrix}, \quad \mathbf{R}_y = \begin{pmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{pmatrix}, \quad \mathbf{R}_z = \begin{pmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Let us recall the formal definition of a rotation. A *rotation* is a linear application which is an isometry (*i.e.*, it preserves the scalar product) which is direct (it does not change the orientation in space).

**Theorem.** A matrix  $\mathbf{R}$  is a rotation matrix if and only if :

$$\mathbf{R}^T \cdot \mathbf{R} = \mathbf{I} \text{ and } \det \mathbf{R} = 1.$$

PROOF. The scalar product is preserved by  $\mathbf{R}$  if, for any  $\mathbf{u}$  and  $\mathbf{v}$  in  $\mathbb{R}^n$ , we have :

$$(\mathbf{R}\mathbf{u})^T \cdot (\mathbf{R}\mathbf{v}) = \mathbf{u}^T \cdot \mathbf{R}^T \mathbf{R} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v}.$$

Therefore  $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ . The symmetries relative to a plane, as well as all the other improper isometries (isometries that change the orientation of space, such as a mirror), also verify the property  $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ . The condition  $\det \mathbf{R} = 1$  allows us to be limited to the isometries which are direct.

### 1.1.2 Lie group

The set of rotation matrices of  $\mathbb{R}^n$  forms a group with respect to the multiplication. It is referred to as a *special orthogonal group* (*special* because  $\det \mathbf{R} = 1$ , *orthogonal* because  $\mathbf{R}^T \cdot \mathbf{R} = \mathbf{I}$ ) and denoted by  $SO(n)$ . It is trivial to check that  $(SO(n), \cdot)$  is a group where  $\mathbf{I}$  is the neutral element. Moreover, the multiplication and the inversion are both smooth. This makes  $SO(n)$  a *Lie group* which is a manifold of the set of matrices  $\mathbb{R}^{n \times n}$ .

The set  $\mathbb{R}^{n \times n}$  of  $n \times n$ -matrices is of dimension  $n^2$ . Since the matrix  $\mathbf{R}^T \cdot \mathbf{R}$  is always symmetric, the matrix equation  $\mathbf{R}^T \cdot \mathbf{R} = \mathbf{I}$  can be decomposed into  $\frac{n(n+1)}{2}$  independent scalar equations. For instance, for  $n = 2$ , we have  $\frac{2(2+1)}{2} = 3$  scalar equations:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix}^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \Leftrightarrow \begin{cases} a^2 + b^2 = 1 \\ ac + bd = 0 \\ c^2 + d^2 = 1 \end{cases}$$

As a consequence, the set  $SO(n)$  forms a manifold of dimension  $d = n^2 - \frac{n(n+1)}{2}$ .

- For  $n = 1$ , we get  $d = 0$ . The set  $SO(1)$  is a singleton which contains a single rotation matrix:  $R = 1$ .
- For  $n = 2$ , we get  $d = 1$ . We need a unique parameter (or angle) to represent the rotations of  $SO(2)$ .
- For  $n = 3$ , we get  $d = 3$ . We need 3 parameters (or angles) to represent  $SO(3)$ .

### 1.1.3 Lie algebra

An *algebra* is an algebraic structure  $(\mathcal{A}, +, \times, \cdot)$  over a body  $\mathbb{K}$ , if (i)  $(\mathcal{A}, +, \cdot)$  is a vector space over  $\mathbb{K}$ ; (ii) the multiplication rule  $\times$  of  $\mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}$  is left- and right-distributive with respect to  $+$  and (iii) for all  $\alpha, \beta \in \mathbb{K}$ , and for all  $x, y \in \mathcal{A}$ ,  $\alpha \cdot x \times \beta \cdot y = (\alpha\beta) \cdot (x \times y)$ . Notice that in general, an algebra is non-commutative ( $x \times y \neq y \times x$ ) and non-associative ( $((x \times y) \times z) \neq x \times (y \times z)$ ). A *Lie algebra*  $(\mathcal{G}, +, [ \cdot, \cdot ], \cdot)$  is a non-commutative and non-associative algebra in which multiplication, denoted by a so-called *Lie bracket*, verifies (i)  $[ \cdot, \cdot ]$  that is bilinear, i.e., linear with respect to each variable ; (ii)  $[x, y] = -[y, x]$  (antisymmetry) and (iii)  $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$  (Jacobi relation).

For Lie groups we can define the associated *Lie algebras*. Lie algebras allow us to consider infinitesimal motions around a given element (*i.e.*, a rotation matrix) in order to use derivatives or differential methods.

Consider the rotation matrix  $\mathbf{I}$  of  $SO(n)$  corresponding to the identity. If we move  $\mathbf{I}$  by adding a small matrix say  $\mathbf{A} \cdot dt$  of  $\mathbb{R}^{n \times n}$ , we generally do not obtain a rotation matrix. We are interested by matrices  $\mathbf{A}$  such that  $\mathbf{I} + \mathbf{A} \cdot dt \in SO(n)$ . We have

$$(\mathbf{I} + \mathbf{A} \cdot dt)^T \cdot (\mathbf{I} + \mathbf{A} \cdot dt) = \mathbf{I}$$

*i.e.*,  $\mathbf{A} \cdot dt + \mathbf{A}^T \cdot dt = \mathbf{0} + o(dt)$ . Therefore,  $\mathbf{A}$  should be skew-symmetric. It means that we are able to move in  $SO(n)$  around  $\mathbf{I}$  by adding infinitesimal skew-symmetric matrices  $\mathbf{A} \cdot dt$  that are not elements of  $SO(n)$ . This corresponds to a new operation in  $SO(n)$  which is not the multiplication we already had. Formally, we define the *Lie algebra* associated to  $SO(n)$  as follows

$$Lie(SO(n)) = \{\mathbf{A} \in \mathbb{R}^{n \times n} \mid \mathbf{I} + \mathbf{A} \cdot dt \in SO(n)\}$$

and it corresponds skew-symmetric matrices of  $\mathbb{R}^{n \times n}$ .

If now we want to move around any matrix  $\mathbf{R}$  of  $SO(n)$ , we generate a rotation matrix around  $\mathbf{I}$  and we transport it to  $\mathbf{R}$ . We get  $\mathbf{R}(\mathbf{I} + \mathbf{A} \cdot dt)$  where  $\mathbf{A}$  is skew-symmetric. It means that we add to  $\mathbf{R}$  the matrix  $\mathbf{R} \cdot \mathbf{A} \cdot dt$ .

In robotics, Lie groups are often used to describe transformations (such as translation or rotations). The Lie algebra corresponds to velocities or equivalently to infinitesimal transformations. Lie group theory is useful, but requires some non trivial mathematical backgrounds that are beyond the scope of this book. We will try to focus on  $SO(3)$  or use more classical tools such as Euler angles, rotation vectors, which are probably less general but sufficient for control purposes.

#### 1.1.4 Rotation vector

If  $\mathbf{R}$  is a rotation matrix depending on time  $t$ , by differentiating the relation  $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ , we get

$$\dot{\mathbf{R}} \cdot \mathbf{R}^T + \mathbf{R} \cdot \dot{\mathbf{R}}^T = \mathbf{0}.$$

Thus, the matrix  $\dot{\mathbf{R}} \cdot \mathbf{R}^T$  is a skew-symmetric matrix (*i.e.*, it satisfies  $\mathbf{A}^T = -\mathbf{A}$  and therefore its diagonal contains only zeros, and for each element of  $\mathbf{A}$ , we have  $a_{ij} = -a_{ji}$ ). We may therefore write, in the case where  $\mathbf{R}$  is of dimension  $3 \times 3$ :

$$\dot{\mathbf{R}} \cdot \mathbf{R}^T = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}. \quad (1.1)$$

The vector  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$  is called the *rotation vector* or *Euler vector* associated with the pair  $(\mathbf{R}, \dot{\mathbf{R}})$ . It must be noted that  $\dot{\mathbf{R}}$  is not a matrix with good properties (such as for instance the fact of being skew-symmetric). On the other hand, the matrix  $\dot{\mathbf{R}} \cdot \mathbf{R}^T$  has the structure of Equation

(1.1) since it allows to be positioned within the coordinate system in which the rotation is performed and this, due to the change of basis performed by  $\mathbf{R}^T$ . We will define the *cross product* between two vectors  $\boldsymbol{\omega}$  and  $\mathbf{x} \in \mathbb{R}^3$  as follows:

$$\boldsymbol{\omega} \wedge \mathbf{x} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \wedge \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_3\omega_y - x_2\omega_z \\ x_1\omega_z - x_3\omega_x \\ x_2\omega_x - x_1\omega_y \end{pmatrix} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

For each vector  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ , we may associate the skew-symmetric matrix:

$$\wedge(\boldsymbol{\omega}) = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

which can be interpreted as the matrix associated with a cross product by the vector  $\boldsymbol{\omega}$ . The matrix  $\wedge(\boldsymbol{\omega})$  is also written  $\boldsymbol{\omega} \wedge$ . It is also called the *small adjoint* of  $\boldsymbol{\omega}$  and denoted by  $\text{ad}(\boldsymbol{\omega})$ . It should not be confused with the large adjoint  $\text{Ad}(\mathbf{R})$  also used in this context, but not here in this book.

**Proposition.** If  $\mathbf{R}(t)$  is a rotation matrix that depends on time, its rotation vector is given by:

$$\boldsymbol{\omega} = \wedge^{-1}(\dot{\mathbf{R}} \cdot \mathbf{R}^T). \quad (1.2)$$

PROOF.– This relation is a direct consequence of Equation (1.1). ■

**Proposition.** If  $\mathbf{R}$  is a rotation matrix in  $\mathbb{R}^3$  and if  $\mathbf{a}$  is a vector of  $\mathbb{R}^3$ , we have:

$$\wedge(\mathbf{R} \cdot \mathbf{a}) = \mathbf{R} \cdot \wedge(\mathbf{a}) \cdot \mathbf{R}^T. \quad (1.3)$$

PROOF.– Let  $\mathbf{x}$  be a vector of  $\mathbb{R}^3$ . We have:

$$\begin{aligned} \wedge(\mathbf{R} \cdot \mathbf{a}) \cdot \mathbf{x} &= (\mathbf{R} \cdot \mathbf{a}) \wedge \mathbf{x} = (\mathbf{R} \cdot \mathbf{a}) \wedge (\mathbf{R} \cdot \mathbf{R}^T \mathbf{x}) \\ &= \mathbf{R} \cdot (\mathbf{a} \wedge \mathbf{R}^T \cdot \mathbf{x}) = \mathbf{R} \cdot \wedge(\mathbf{a}) \cdot \mathbf{R}^T \cdot \mathbf{x}. \quad \blacksquare \end{aligned}$$

**Proposition.** (duality). We have:

$$\mathbf{R}^T \dot{\mathbf{R}} = \wedge(\mathbf{R}^T \boldsymbol{\omega}). \quad (1.4)$$

This relation expresses the fact that the matrix  $\mathbf{R}^T \dot{\mathbf{R}}$  is associated with the rotation vector  $\boldsymbol{\omega}$ , associated with  $\mathbf{R}(t)$  but expressed in the coordinate system associated with  $\mathbf{R}$  whereas  $\dot{\mathbf{R}} \cdot \mathbf{R}^T$  is associated to the same vector, but this time expressed in the coordinate system of the standard basis.

PROOF.– We have:

$$\mathbf{R}^T \dot{\mathbf{R}} = \mathbf{R}^T (\dot{\mathbf{R}} \cdot \mathbf{R}^T) \mathbf{R} \stackrel{(1.2)}{=} \mathbf{R}^T \cdot \wedge(\boldsymbol{\omega}) \cdot \mathbf{R} \stackrel{(1.3)}{=} \wedge(\mathbf{R}^T \boldsymbol{\omega}). \quad \blacksquare$$

### 1.1.5 Rodrigues rotation formulas

**Matrix exponential.** Given a square matrix  $\mathbf{M}$  of dimension  $n$ , its exponential can be defined as:

$$e^{\mathbf{M}} = \mathbf{I}_n + \mathbf{M} + \frac{1}{2!}\mathbf{M}^2 + \frac{1}{3!}\mathbf{M}^3 + \cdots = \sum_{i=0}^{\infty} \frac{1}{i!}\mathbf{M}^i$$

where  $\mathbf{I}_n$  is the identity matrix of dimension  $n$ . It is clear that  $e^{\mathbf{M}}$  is of the same dimension as  $\mathbf{M}$ . Here are some of the important properties concerning the exponentials of matrices. If  $\mathbf{0}_n$  is the zero matrix of  $n \times n$  and if  $\mathbf{M}$  and  $\mathbf{N}$  are two matrices  $n \times n$ , then:

$$\begin{aligned} e^{\mathbf{0}_n} &= \mathbf{I}_n \\ e^{\mathbf{M}} \cdot e^{\mathbf{N}} &= e^{\mathbf{M}+\mathbf{N}} \text{ (if the matrices commute)} \\ \frac{d}{dt} (e^{\mathbf{Mt}}) &= \mathbf{M}e^{\mathbf{Mt}} \end{aligned}$$

**Matrix logarithm.** Given a matrix  $\mathbf{M}$  of dimension  $n$ , the matrix  $\mathbf{L}$  is said to be a matrix logarithm of  $\mathbf{M}$  if  $e^{\mathbf{L}} = \mathbf{M}$ . As for complex numbers, the exponential function is not a one-to-one function and matrices may have more than one logarithm. Using power series, we define the logarithm of a square matrix as

$$\log \mathbf{M} = \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} (\mathbf{M} - \mathbf{I}_n)^i.$$

The sum is convergent if  $\mathbf{M}$  is close to identity.

**Rodrigues formulas.** A rotation matrix  $\mathbf{R}$  has an axis represented by a unit vector  $\mathbf{n}$  and an angle  $\alpha$  with respect to this axis. From  $\mathbf{n}$  and  $\alpha$  we can also generate the matrix  $\mathbf{R}$ . The link  $\mathbf{R} \leftrightarrow (\mathbf{n}, \alpha)$  is made by the following *Rodrigues formulas*

$$\begin{aligned} (i) \quad \mathbf{R} &= e^{\alpha \mathbf{n} \wedge} \\ (ii) \quad \alpha \mathbf{n} &= \log \mathbf{R} \end{aligned} \tag{1.5}$$

The first equation (i) will be shown in Exercise 1.4 and (ii) is the reciprocal of (i). In these formulas,  $\alpha \mathbf{n} \wedge$  is a notation to represent the matrix  $\wedge(\alpha \mathbf{n})$ . As it will be shown in the exercise,  $\mathbf{R} = e^{\alpha \mathbf{n} \wedge}$  is a rotation matrix and  $\mathbf{n}$  is an eigenvector associated with the eigenvalue 1 of  $\mathbf{R}$ .

**Axis and angle of a rotation matrix.** Given a rotation matrix  $\mathbf{R}$ , the axis can be represented by normalized eigen vector  $\mathbf{n}$  associated to the eigen value  $\lambda = 1$  (two of them can be found, but we need one). To compute  $\alpha$  and  $\mathbf{n}$ , we may use the relation  $\alpha \mathbf{n} = \log \mathbf{R}$ , which works well if  $\mathbf{R}$  is close to identity. In the general case, it is more suitable to use the following proposition.

**Proposition.** Given a rotation matrix  $\mathbf{R}$ , we have  $\mathbf{R} = e^{\alpha \mathbf{n} \wedge}$  where

$$\begin{aligned} \alpha &= \frac{\operatorname{acos}(\frac{\operatorname{tr}(\mathbf{R})-1}{2})}{2} \\ \mathbf{n} &= \frac{1}{2\sin\alpha} \cdot \wedge^{-1} (\mathbf{R} - \mathbf{R}^T) \end{aligned} \tag{1.6}$$

PROOF.– Take a rotation matrix  $\mathbf{R} = e^{\alpha \mathbf{n} \wedge}$  with eigen values  $1, \lambda_1, \lambda_2$  and eigen vectors  $\mathbf{n}, \mathbf{v}_1, \mathbf{v}_2$ . Consider the generalized polynomial  $f(x) = x - x^{-1}$ , where  $x$  is the indeterminate. From the

correspondence theorem of eigen values/vectors, the eigen values of  $f(\mathbf{R}) = \mathbf{R} - \mathbf{R}^{-1} = \mathbf{R} - \mathbf{R}^T$  are  $f(1) = 0, f(\lambda_2), f(\lambda_3)$  and the eigen vectors are still  $\mathbf{n}, \mathbf{v}_1, \mathbf{v}_2$ . Now,

- |        |                                               |                                                                                                          |         |
|--------|-----------------------------------------------|----------------------------------------------------------------------------------------------------------|---------|
| $(i)$  | $f(\mathbf{R})$ is skew symmetric             | $(f^T(\mathbf{R}) + f(\mathbf{R}) = \mathbf{R}^T - \mathbf{R} + \mathbf{R} - \mathbf{R}^T = \mathbf{0})$ | $(1.7)$ |
| $(ii)$ | $f(\mathbf{R}) \cdot \mathbf{n} = \mathbf{0}$ | $(\mathbf{n} \text{ is the eigen vector of } f(\mathbf{R}) \text{ associated to } 0)$                    |         |

Thus, the two matrices  $f(\mathbf{R})$  and  $(\mathbf{n} \wedge)$  are proportional. As a consequence  $\wedge^{-1}(f(\mathbf{R}))$  provides us the axis of  $\mathbf{R}$ . To find the angle  $\alpha$ , we use the property that the trace of a matrix is *similarity-invariant*<sup>1</sup>, which means that for any invertible matrix  $\mathbf{P}$ , we have  $\text{tr}(\mathbf{R}) = \text{tr}(\mathbf{P}^{-1} \cdot \mathbf{R} \cdot \mathbf{P})$ . Take  $\mathbf{P}$  as the rotation matrix with transforms  $\mathbf{R}$  into a rotation along the first axis. We get

$$\text{tr}(\mathbf{R}) = \text{tr} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} = 1 + 2 \cos \alpha,$$

which provides us the angle of the rotation. Exercise ?? provides a geometrical proof of this proposition.

### 1.1.6 Coordinate system change

Let  $\mathcal{R}_0 : (\mathbf{o}_0, \mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0)$  and  $\mathcal{R}_1 : (\mathbf{o}_1, \mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1)$  be two coordinate systems and let  $\mathbf{u}$  be a vector of  $\mathbb{R}^3$  (refer to Figure 1.3). We have the following relation:

$$\begin{aligned} \mathbf{u} &= x_0 \mathbf{i}_0 + y_0 \mathbf{j}_0 + z_0 \mathbf{k}_0 \\ &= x_1 \mathbf{i}_1 + y_1 \mathbf{j}_1 + z_1 \mathbf{k}_1 \end{aligned}$$

where  $(x_0, y_0, z_0)$  and  $(x_1, y_1, z_1)$  are the coordinates of  $\mathbf{u}$  in  $\mathcal{R}_0$  and  $\mathcal{R}_1$ , respectively.

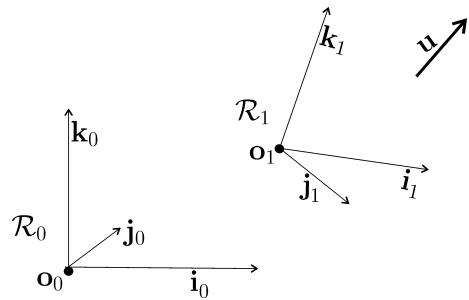


Figure 1.3: Changing the coordinate system  $\mathcal{R}_0$  to the system  $\mathcal{R}_1$

Thus, for any vector  $\mathbf{v}$  we have:

$$\langle x_0 \mathbf{i}_0 + y_0 \mathbf{j}_0 + z_0 \mathbf{k}_0, \mathbf{v} \rangle = \langle x_1 \mathbf{i}_1 + y_1 \mathbf{j}_1 + z_1 \mathbf{k}_1, \mathbf{v} \rangle.$$

---

<sup>1</sup> $\text{tr}((\mathbf{P}^{-1} \cdot \mathbf{R}) \cdot \mathbf{P}) = \text{tr}(\mathbf{P} \cdot (\mathbf{P}^{-1} \cdot \mathbf{R})) = \text{tr}(\mathbf{R})$

By taking respectively  $\mathbf{v} = \mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0$ , we obtain the following three relations:

$$\begin{cases} \langle x_0 \mathbf{i}_0 + y_0 \mathbf{j}_0 + z_0 \mathbf{k}_0, \mathbf{i}_0 \rangle = \langle x_1 \mathbf{i}_1 + y_1 \mathbf{j}_1 + z_1 \mathbf{k}_1, \mathbf{i}_0 \rangle \\ \langle x_0 \mathbf{i}_0 + y_0 \mathbf{j}_0 + z_0 \mathbf{k}_0, \mathbf{j}_0 \rangle = \langle x_1 \mathbf{i}_1 + y_1 \mathbf{j}_1 + z_1 \mathbf{k}_1, \mathbf{j}_0 \rangle \\ \langle x_0 \mathbf{i}_0 + y_0 \mathbf{j}_0 + z_0 \mathbf{k}_0, \mathbf{k}_0 \rangle = \langle x_1 \mathbf{i}_1 + y_1 \mathbf{j}_1 + z_1 \mathbf{k}_1, \mathbf{k}_0 \rangle \end{cases}$$

However, since the basis  $(\mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0)$  of  $\mathcal{R}_0$  is orthonormal,  $\langle \mathbf{i}_0, \mathbf{i}_0 \rangle = \langle \mathbf{j}_0, \mathbf{j}_0 \rangle = \langle \mathbf{k}_0, \mathbf{k}_0 \rangle = 1$  and  $\langle \mathbf{i}_0, \mathbf{j}_0 \rangle = \langle \mathbf{j}_0, \mathbf{k}_0 \rangle = \langle \mathbf{i}_0, \mathbf{k}_0 \rangle = 0$ . Thus, these three relations become:

$$\begin{cases} x_0 = x_1 \cdot \langle \mathbf{i}_1, \mathbf{i}_0 \rangle + y_1 \cdot \langle \mathbf{j}_1, \mathbf{i}_0 \rangle + z_1 \cdot \langle \mathbf{k}_1, \mathbf{i}_0 \rangle \\ y_0 = x_1 \cdot \langle \mathbf{i}_1, \mathbf{j}_0 \rangle + y_1 \cdot \langle \mathbf{j}_1, \mathbf{j}_0 \rangle + z_1 \cdot \langle \mathbf{k}_1, \mathbf{j}_0 \rangle \\ z_0 = x_1 \cdot \langle \mathbf{i}_1, \mathbf{k}_0 \rangle + y_1 \cdot \langle \mathbf{j}_1, \mathbf{k}_0 \rangle + z_1 \cdot \langle \mathbf{k}_1, \mathbf{k}_0 \rangle \end{cases}$$

Or in matrix form:

$$\underbrace{\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}}_{=\mathbf{u}|_{\mathcal{R}_0}} = \underbrace{\begin{pmatrix} \langle \mathbf{i}_1, \mathbf{i}_0 \rangle & \langle \mathbf{j}_1, \mathbf{i}_0 \rangle & \langle \mathbf{k}_1, \mathbf{i}_0 \rangle \\ \langle \mathbf{i}_1, \mathbf{j}_0 \rangle & \langle \mathbf{j}_1, \mathbf{j}_0 \rangle & \langle \mathbf{k}_1, \mathbf{j}_0 \rangle \\ \langle \mathbf{i}_1, \mathbf{k}_0 \rangle & \langle \mathbf{j}_1, \mathbf{k}_0 \rangle & \langle \mathbf{k}_1, \mathbf{k}_0 \rangle \end{pmatrix}}_{=\mathbf{R}_{\mathcal{R}_0}^{\mathcal{R}_1}} \cdot \underbrace{\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}}_{=\mathbf{u}|_{\mathcal{R}_1}} \quad (1.8)$$

We can see a rotation matrix  $\mathbf{R}_{\mathcal{R}_0}^{\mathcal{R}_1}$  whose columns are the coordinates of  $\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1$  expressed in the absolute system  $\mathcal{R}_0$ . Thus

$$\mathbf{R}_{\mathcal{R}_0}^{\mathcal{R}_1} = \left( \begin{array}{c|c|c} \mathbf{i}_1|_{\mathcal{R}_0} & \mathbf{j}_1|_{\mathcal{R}_0} & \mathbf{k}_1|_{\mathcal{R}_0} \end{array} \right)$$

This matrix depends on time and links the frame  $\mathcal{R}_1$  to  $\mathcal{R}_0$ . The matrix  $\mathbf{R}_{\mathcal{R}_0}^{\mathcal{R}_1}$  is often referred to as a *direction cosine matrix* since its components involve the direction cosines of the basis vectors of the two coordinate systems. Likewise, if we would have several systems  $\mathcal{R}_0, \dots, \mathcal{R}_n$  (see Figure 1.4), we would have:

$$\mathbf{u}|_{\mathcal{R}_0} = \mathbf{R}_{\mathcal{R}_0}^{\mathcal{R}_1} \cdot \mathbf{R}_{\mathcal{R}_1}^{\mathcal{R}_2} \cdot \dots \cdot \mathbf{R}_{\mathcal{R}_{n-1}}^{\mathcal{R}_n} \cdot \mathbf{u}|_{\mathcal{R}_n}.$$

**Dead Reckoning.** Let us consider the situation of a robot moving in a three-dimensional environment. Let us call  $\mathcal{R}_0 : (\mathbf{o}_0, \mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0)$  its reference frame (for example, the frame of the robot at an initial time). The position of the robot is represented by the vector  $\mathbf{p}(t)$  expressed in  $\mathcal{R}_0$  and its attitude (*i.e.*, its orientation) by the rotation matrix  $\mathbf{R}(t)$  which represents the coordinates of the vectors  $\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1$  of the coordinate system  $\mathcal{R}_1$  of the robot expressed in the coordinate system  $\mathcal{R}_0$ , at time  $t$ . It follows that:

$$\mathbf{R}(t) = \left( \begin{array}{c|c|c} \mathbf{i}_1|_{\mathcal{R}_0} & \mathbf{j}_1|_{\mathcal{R}_0} & \mathbf{k}_1|_{\mathcal{R}_0} \end{array} \right) = \mathbf{R}_{\mathcal{R}_0}^{\mathcal{R}_1}(t).$$

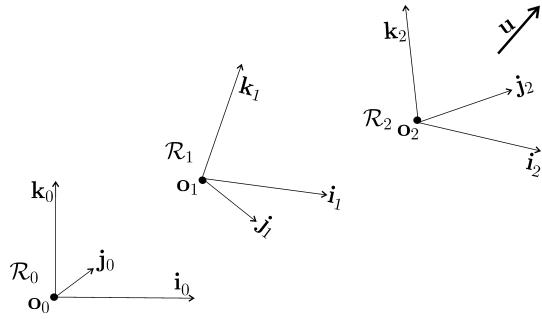


Figure 1.4: Composition of rotations

This matrix can be returned by a precise attitude unit positioned on the robot. If the robot is also equipped with a *Doppler Velocity Log* (or DVL) which provides it with its speed vector  $\mathbf{v}_r$  relative to the ground or the seabed, expressed in the coordinate system  $\mathcal{R}_1$  of the robot, then the speed vector  $\mathbf{v}$  of the robot satisfies:

$$\underbrace{\dot{\mathbf{p}}(t)}_{\mathbf{v}|_{\mathcal{R}_0}} \stackrel{(1.8)}{=} \underbrace{\mathbf{R}(t)}_{\mathbf{R}^{\mathcal{R}_0}} \cdot \underbrace{\mathbf{v}_r(t)}_{\mathbf{v}|_{\mathcal{R}_1}}$$

Or equivalently:

$$\dot{\mathbf{p}}(t) = \mathbf{R}(t) \cdot \mathbf{v}_r(t). \quad (1.9)$$

*Dead reckoning* consists of integrating this state equation from the knowledge of  $\mathbf{R}(t)$  and  $\mathbf{v}_r(t)$ .

## 1.2 Euler angles

### 1.2.1 Definition

In related literature, the angles proposed by Euler in 1770 to represent the orientation of solid bodies in space are not uniquely defined. We mainly distinguish between the roll-yaw-roll, roll-pitch-roll and roll-pitch-yaw formulations. It is the latter that we will choose since it is imposed in the mobile robotics language. Within this roll-pitch-yaw formulation, the Euler angles are sometimes referred to as *Cardan angles*. Any rotation matrix of  $\mathbb{R}^3$  can be expressed in the form of the product of three matrices as follows:

$$\mathbf{R}(\varphi, \theta, \psi) = \underbrace{\begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{R}_\psi} \cdot \underbrace{\begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}}_{\mathbf{R}_\theta} \cdot \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix}}_{\mathbf{R}_\varphi}$$

or, in developed form:

$$\begin{pmatrix} \cos \theta \cos \psi & -\cos \varphi \sin \psi + \sin \theta \cos \psi \sin \varphi & \sin \psi \sin \varphi + \sin \theta \cos \psi \cos \varphi \\ \cos \theta \sin \psi & \cos \psi \cos \varphi + \sin \theta \sin \psi \sin \varphi & -\cos \psi \sin \varphi + \sin \theta \cos \varphi \sin \psi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{pmatrix} \quad (1.10)$$

$\overbrace{\phantom{\begin{pmatrix} \cos \theta \cos \psi & -\cos \varphi \sin \psi + \sin \theta \cos \psi \sin \varphi & \sin \psi \sin \varphi + \sin \theta \cos \psi \cos \varphi \\ \cos \theta \sin \psi & \cos \psi \cos \varphi + \sin \theta \sin \psi \sin \varphi & -\cos \psi \sin \varphi + \sin \theta \cos \varphi \sin \psi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{pmatrix}}^{\mathbf{i}_1|_{\mathcal{R}_0}}$

$\overbrace{\phantom{\begin{pmatrix} \cos \theta \cos \psi & -\cos \varphi \sin \psi + \sin \theta \cos \psi \sin \varphi & \sin \psi \sin \varphi + \sin \theta \cos \psi \cos \varphi \\ \cos \theta \sin \psi & \cos \psi \cos \varphi + \sin \theta \sin \psi \sin \varphi & -\cos \psi \sin \varphi + \sin \theta \cos \varphi \sin \psi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{pmatrix}}^{\mathbf{j}_1|_{\mathcal{R}_0}}$

$\overbrace{\phantom{\begin{pmatrix} \cos \theta \cos \psi & -\cos \varphi \sin \psi + \sin \theta \cos \psi \sin \varphi & \sin \psi \sin \varphi + \sin \theta \cos \psi \cos \varphi \\ \cos \theta \sin \psi & \cos \psi \cos \varphi + \sin \theta \sin \psi \sin \varphi & -\cos \psi \sin \varphi + \sin \theta \cos \varphi \sin \psi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{pmatrix}}^{\mathbf{k}_1|_{\mathcal{R}_0}}$

The angles  $\varphi, \theta, \psi$  are the *Euler angles* and are respectively called the *bank*, the *elevation* and the *heading*. The terms *roll*, *pitch*, *yaw* are often employed, although they correspond, respectively, to variations of bank, elevation and heading.

**Gimbal lock.** When  $\theta = \frac{\pi}{2}$  (the same effect happens as soon as  $\cos \theta = 0$ ), we have

$$\begin{aligned} \mathbf{R}(\varphi, \theta, \psi) &= \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \underbrace{\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}}_{= \begin{pmatrix} 0 & \sin \varphi & \cos \varphi \\ 0 & \cos \varphi & -\sin \varphi \\ -1 & 0 & 0 \end{pmatrix}} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \\ &= \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \underbrace{\begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & -1 \end{pmatrix}}_{= \begin{pmatrix} \cos(\psi - \varphi) & -\sin(\psi - \varphi) & 0 \\ \sin(\psi - \varphi) & \cos(\psi - \varphi) & 0 \\ 0 & 0 & -1 \end{pmatrix}} \cdot \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\psi - \varphi) & -\sin(\psi - \varphi) & 0 \\ \sin(\psi - \varphi) & \cos(\psi - \varphi) & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{aligned}$$

and thus,

$$\frac{d\mathbf{R}}{d\psi} = -\frac{d\mathbf{R}}{d\varphi}.$$

This corresponds to a singularity which tells us that when  $\theta = \frac{\pi}{2}$ , we cannot move on the manifold of rotation matrices  $SO(3)$ , in all directions using the Euler angles. Equivalently, this means that some trajectories  $\mathbf{R}(t)$  cannot be followed by Euler angles.

**Rotation matrix to Euler angles.** Given a rotation matrix  $\mathbf{R}$ , we can easily find the three Euler angles by solving, following Equation (1.10), the equations:

$$\begin{cases} -\sin \theta = r_{31} \\ \cos \theta \sin \varphi = r_{32} & \cos \theta \cos \varphi = r_{33} \\ \cos \theta \cos \psi = r_{11} & \cos \theta \sin \psi = r_{21} \end{cases}$$

By imposing  $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ ,  $\varphi \in [-\pi, \pi]$ ,  $\psi \in [-\pi, \pi]$ , we find:

$$\theta = -\arcsin r_{31}, \varphi = \text{atan2}(r_{32}, r_{33}) \text{ et } \psi = \text{atan2}(r_{21}, r_{11}).$$

Here atan2 is the two-argument arctangent function defined by

$$\alpha = \text{atan2}(y, x) \Leftrightarrow \alpha \in ]-\pi, \pi] \text{ and } \exists r > 0 \mid \begin{cases} x = r \cos \alpha \\ y = r \sin \alpha \end{cases} \quad (1.11)$$

### 1.2.2 Rotation vector of a moving Euler matrix

Let us consider a solid body moving in a coordinate system  $\mathcal{R}_0$  and a coordinate system  $\mathcal{R}_1$  attached to this body (refer to Figure 1.5). The conventions chosen here are those of the SNAME (*Society of Naval and Marine Engineers*). The two coordinate systems are assumed to be orthonormal. Let  $\mathbf{R}(t) = \mathbf{R}(\psi(t), \theta(t), \varphi(t))$  be the rotation matrix that links the two systems. We need to find the instantaneous rotation vector  $\boldsymbol{\omega}$  of the solid body relative to  $\mathcal{R}_0$  in function of  $\varphi, \theta, \psi, \dot{\varphi}, \dot{\theta}, \dot{\psi}$ . We have:

$$\begin{aligned} & \dot{\mathbf{R}} \cdot \mathbf{R}^T \\ & \stackrel{(1.10)}{=} \frac{d}{dt} (\mathbf{R}_\psi \cdot \mathbf{R}_\theta \cdot \mathbf{R}_\varphi) \cdot \mathbf{R}_\varphi^T \cdot \mathbf{R}_\theta^T \cdot \mathbf{R}_\psi^T \\ & = \left( \dot{\mathbf{R}}_\psi \cdot \mathbf{R}_\theta \cdot \mathbf{R}_\varphi + \mathbf{R}_\psi \cdot \dot{\mathbf{R}}_\theta \cdot \mathbf{R}_\varphi + \mathbf{R}_\psi \cdot \mathbf{R}_\theta \cdot \dot{\mathbf{R}}_\varphi \right) \cdot \mathbf{R}_\varphi^T \cdot \mathbf{R}_\theta^T \cdot \mathbf{R}_\psi^T \\ & \stackrel{(1.2)}{=} \underbrace{\dot{\mathbf{R}}_\psi \cdot \mathbf{R}_\psi^T}_{=\wedge(\dot{\psi}\mathbf{k})} + \mathbf{R}_\psi \cdot \underbrace{\dot{\mathbf{R}}_\theta \cdot \mathbf{R}_\theta^T}_{=\wedge(\dot{\theta}\mathbf{j})} \cdot \mathbf{R}_\psi^T + \mathbf{R}_\psi \cdot \mathbf{R}_\theta \cdot \underbrace{\dot{\mathbf{R}}_\varphi \cdot \mathbf{R}_\varphi^T}_{=\wedge(\dot{\varphi}\mathbf{i})} \cdot \mathbf{R}_\theta^T \cdot \mathbf{R}_\psi^T \\ & \stackrel{(1.3)}{=} \wedge(\dot{\psi}\mathbf{k}) + \wedge(\dot{\theta} \cdot \mathbf{R}_\psi \cdot \mathbf{j}) + \wedge(\dot{\varphi} \cdot \mathbf{R}_\psi \cdot \mathbf{R}_\theta \cdot \mathbf{i}) \\ & = \wedge(\dot{\psi}\mathbf{k} + \dot{\theta} \cdot \mathbf{R}_\psi \cdot \mathbf{j} + \dot{\varphi} \cdot \mathbf{R}_\psi \cdot \mathbf{R}_\theta \cdot \mathbf{i}). \end{aligned}$$

Thus

$$\begin{aligned} \boldsymbol{\omega}|_{\mathcal{R}_0} & \stackrel{(1.2)}{=} \wedge^{-1} (\dot{\mathbf{R}} \cdot \mathbf{R}^T) \\ & = \dot{\psi} \cdot \mathbf{k} + \dot{\theta} \cdot \mathbf{R}_\psi \cdot \mathbf{j} + \dot{\varphi} \cdot \mathbf{R}_\psi \cdot \mathbf{R}_\theta \cdot \mathbf{i}. \end{aligned}$$

After having calculated the quantities  $\mathbf{k}$ ,  $\mathbf{R}_\psi \mathbf{j}$  and  $\mathbf{R}_\psi \cdot \mathbf{R}_\theta \cdot \mathbf{i}$  in the coordinate system  $\mathcal{R}_0$ , we get the result:

$$\boldsymbol{\omega}|_{\mathcal{R}_0} = \begin{pmatrix} \cos \theta \cos \psi & -\sin \psi & 0 \\ \cos \theta \sin \psi & \cos \psi & 0 \\ -\sin \theta & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}. \quad (1.12)$$

This matrix is singular if and only if  $\cos \theta = 0$ . We will therefore take care to never have an elevation  $\theta$  equal to  $\pm \frac{\pi}{2}$ , to avoid the gimbal lock.

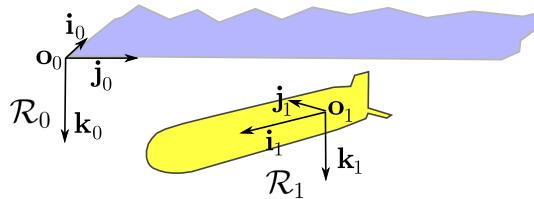


Figure 1.5: The coordinate system  $\mathcal{R}_1 : (\mathbf{o}_1, \mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1)$  attached to the robot

### 1.3 Inertial unit

Assume that we are enclosed inside a box near to the Earth without any possibility to see the environment outside. Inside the box we may perform some inertial experiments such as observing bodies translating or rotating. These experiments allow us to measure indirectly the tangential accelerations and the angular speed of the box in its own frame. The principle of an inertial unit is to estimate the pose (position and orientation) of the box from these inertial measurements only (called *proprioceptive measurements*), assuming that the initial pose is known.

For this purpose, we will describe the motion of our box by a kinematic model the inputs of which are the tangential accelerations and the angular speeds these are directly measurable in the coordinate system of the box. The state vector is composed of the vector  $\mathbf{p} = (p_x, p_y, p_z)$  that gives the coordinates of the center of the box expressed in the absolute inertial coordinate system  $\mathcal{R}_0$ , the three Euler angles  $(\varphi, \theta, \psi)$  and the speed vector  $\mathbf{v}_r$  of the box expressed in its own coordinate system  $\mathcal{R}_1$ . The inputs of the system are for one the acceleration  $\mathbf{a}_r = \mathbf{a}_{\mathcal{R}_1}$  of the center of the box also given in  $\mathcal{R}_1$  and second, the vector  $\boldsymbol{\omega}_r = \boldsymbol{\omega}_{\mathcal{R}_1/\mathcal{R}_0|\mathcal{R}_1} = (\omega_x, \omega_y, \omega_z)$  corresponding to the rotation vector of the box relative to  $\mathcal{R}_0$  expressed in  $\mathcal{R}_1$ . We have indeed to express  $\mathbf{a}, \boldsymbol{\omega}$  in the coordinate system of the box since these quantities are generally measured *via* the sensors attached on it. The first state equation is:

$$\dot{\mathbf{p}} \stackrel{(1.9)}{=} \mathbf{R}(\varphi, \theta, \psi) \cdot \mathbf{v}_r.$$

Let us differentiate this equation. We obtain:

$$\ddot{\mathbf{p}} = \dot{\mathbf{R}} \cdot \mathbf{v}_r + \mathbf{R} \cdot \dot{\mathbf{v}}_r$$

with  $\mathbf{R} = \mathbf{R}(\varphi, \theta, \psi)$ . From this equation, we isolate  $\dot{\mathbf{v}}_r$  to get

$$\dot{\mathbf{v}}_r = \underbrace{\mathbf{R}^T \cdot \ddot{\mathbf{p}}}_{\mathbf{a}_r} - \mathbf{R}^T \dot{\mathbf{R}} \cdot \mathbf{v}_r \stackrel{(1.4)}{=} \mathbf{a}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r.$$

which constitutes the second state equation. Figure 1.6 shows a situation where a robot has a constant speed and follows a circle. The speed vector  $\mathbf{v}_r$  is a constant whereas  $\mathbf{a}_r$  is different from zero.

We also need to express  $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$  in function of the state variables. From Equation (1.12), the relation

$$\boldsymbol{\omega}_{|\mathcal{R}_0} = \mathbf{R}(\varphi, \theta, \psi) \cdot \boldsymbol{\omega}_{|\mathcal{R}_1}$$

becomes

$$\begin{pmatrix} \cos \theta \cos \psi & -\sin \psi & 0 \\ \cos \theta \sin \psi & \cos \psi & 0 \\ -\sin \theta & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \mathbf{R}(\varphi, \theta, \psi) \cdot \boldsymbol{\omega}_r.$$

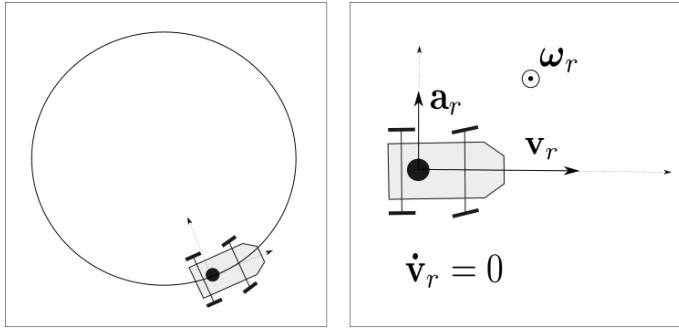


Figure 1.6: Illustration of the formula  $\dot{\mathbf{v}}_r = \mathbf{a}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r$  when  $\dot{\mathbf{v}}_r = 0$ . Left: the robot follows a circle; right: representation of the vectors in the robot frame.

By isolating in this expression the vector  $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$ , we obtain the third state equation. By bringing together the three state equations, we obtain the *navigation mechanization equations* [FAR 08]:

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R}(\varphi, \theta, \psi) \cdot \mathbf{v}_r \\ \dot{\mathbf{v}}_r &= \mathbf{a}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} &= \begin{pmatrix} 1 & \tan \theta \sin \varphi & \tan \theta \cos \varphi \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \cdot \boldsymbol{\omega}_r \end{cases} \quad (1.13)$$

**On a horizontal plane.** If we move on a horizontal plane, we have  $\varphi = \theta = 0$ . The last equation gives us  $\dot{\psi} = \omega_{r3}$ ,  $\dot{\theta} = \omega_{r2}$  and  $\dot{\varphi} = \omega_{r1}$ . In such a case, there is a perfect correspondence between the components of  $\boldsymbol{\omega}_r$  and the differentials of the Euler angles. There are singular cases, for instance when  $\theta = \frac{\pi}{2}$  (this is the case when the box points upwards), in which the differentials of the Euler angles cannot be defined. Using the rotation vector is often preferred since it does not have such singularities.

**Dead reckoning.** For dead reckoning (*i.e.*, without external sensors) there are generally extremely precise laser gyroscopes (around 0.001 deg/s.). These make use of the Sagnac effect (in a circular optical fiber turning around itself, the time taken by light to travel an entire round-trip depends on the path direction). Using three fibers, these gyroscopes generate the vector  $\boldsymbol{\omega}_r = (\omega_x, \omega_y, \omega_z)$ . There are also accelerometers capable of measuring the acceleration  $\mathbf{a}_r$  with a very high degree of precision. In pure inertial mode, we determine our position by differentiating Equations (1.13) only using the acceleration  $\mathbf{a}_r$  and the rotation speed  $\boldsymbol{\omega}_r$ , both expressed in the coordinate system of the box. In the case where we are measuring the quantity  $\mathbf{v}_r$  (also expressed in the frame of the inertial unit attached to the robot) with a *Doppler velocity log*, we only need to integrate the first and the last of these three equations. Finally, when the robot is a correctly ballasted submarine or a terrestrial robot moving on a relatively plane ground, we know *a priori* that on average the bank and the elevation are equal to zero. We may therefore incorporate this information through a Kalman filter in order to limit the drift in positioning. An efficient inertial unit integrates an amalgamation of all the available information.

**Inertial unit.** A pure inertial unit (without hybridization and without taking into account

Earth's gravity) represents the robot by the kinematic model of Figure 1.7, which itself uses the state equations given in (1.13). This system is written in the form  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$  where  $\mathbf{u} = (\mathbf{a}_r, \boldsymbol{\omega}_r)$  is the vector of the measured inertial inputs (accelerations and rotation speeds viewed by an observer on the ground, but expressed in the frame of the robot) and  $\mathbf{x} = (\mathbf{p}, \mathbf{v}_r, \varphi, \theta, \psi)$  is the state vector. We may use a numerical integration method such as the Euler method

$$\mathbf{x}(t + dt) = \mathbf{x}(t) + dt \cdot \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$$

or a Runge-Kutta integration scheme:

$$\mathbf{x}(t + dt) \simeq \mathbf{x}(t) + dt \cdot \left( \frac{\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))}{4} + \frac{3}{4}\mathbf{f}(\mathbf{x}(t) + \frac{2dt}{3}\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{u}(t + \frac{2}{3}dt)) \right)$$

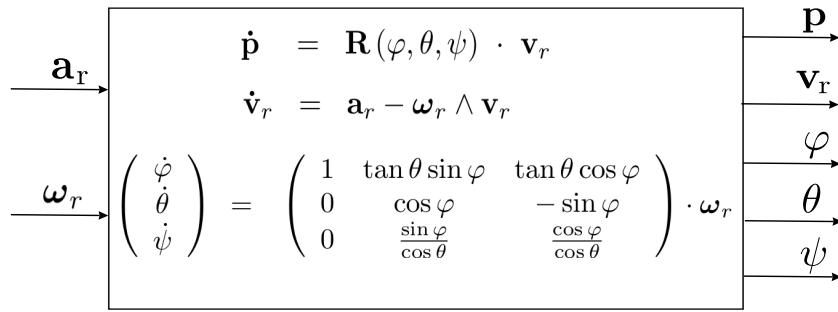


Figure 1.7: Navigation mechanization equations

**With gravity.** In case where there exists a gravity  $\mathbf{g}(\mathbf{p})$  which depends on  $\mathbf{p}$ , the actual acceleration  $\mathbf{a}_r$  corresponds to the gravity plus the measured acceleration  $\mathbf{a}_r^{mes}$ . In this case, the inertial unit can be described by the following state equations

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R}(\varphi, \theta, \psi) \cdot \mathbf{v}_r \\ \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} &= \begin{pmatrix} 1 & \tan \theta \sin \varphi & \tan \theta \cos \varphi \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \cdot \boldsymbol{\omega}_r \\ \dot{\mathbf{v}}_r &= \mathbf{R}^T(\varphi, \theta, \psi) \cdot \mathbf{g}(\mathbf{p}) + \mathbf{a}_r^{mes} - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \end{cases} \quad (1.14)$$

In this case, the input of the system are  $\mathbf{a}_r^{mes}$  and  $\boldsymbol{\omega}_r$  which are obtained by the accelerometers and the gyroscope. To integrate this state equation for localization purpose, we need to know the initial state and a *gravity map*  $\mathbf{g}(\mathbf{p})$ .

**Using rotation matrix instead of Euler angles.** From Equation (1.4), we have  $\dot{\mathbf{R}} = \mathbf{R} \cdot \wedge(\mathbf{R}^T \boldsymbol{\omega})$ . Thus, the state equations (1.14) can be written without any Euler angles as

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \cdot \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R} \cdot (\boldsymbol{\omega}_r \wedge) \\ \dot{\mathbf{v}}_r &= \mathbf{R}^T \cdot \mathbf{g}(\mathbf{p}) + \mathbf{a}_r^{mes} - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \end{cases} \quad (1.15)$$

The main advantage of this representation is that we do not have the singularity that exists in (1.14) for  $\cos \theta = 0$ . But instead, we have redundancies, since  $\mathbf{R}$  of dimension 9 replaces the 3 Euler angles.

Again, we can use a Runge Kutta integration. Now, for the integration of  $\mathbf{R}(t)$ , it is more accurate to compute

$$\mathbf{R}(t + dt) = \mathbf{R}(t) \cdot \exp(dt \cdot \boldsymbol{\omega}_r(t) \wedge).$$

For numerical reasons, when we integrate for a long time, the matrix  $\mathbf{R}(t)$  may lose the property  $\mathbf{R} \cdot \mathbf{R}^T = \mathbf{I}$ . To avoid this, at each iteration, a renormalization is needed, *i.e.*, the current matrix  $\mathbf{R}$  should be projected on to  $SO(3)$ . A possibility is to use a QR factorization. A QR-decomposition of a square matrix  $\mathbf{M}$  provides two matrices  $\mathbf{Q}, \mathbf{R}$  such that  $\mathbf{M} = \mathbf{Q} \cdot \mathbf{R}$  where  $\mathbf{Q}$  is a rotation matrix and  $\mathbf{R}$  is triangular. When  $\mathbf{M}$  is almost a rotation matrix, then  $\mathbf{R}$  is almost diagonal and also almost a rotation, *i.e.*, on the diagonal of  $\mathbf{R}$  the entries are approximately  $\pm 1$  and outside the entries are almost zeros. The following PYTHON code performs the projection of  $\mathbf{M}$  on  $SO(3)$  and generates the rotation matrix  $\mathbf{M}_2$ .

```
Q,R = numpy.linalg.qr(M)
v=diag(sign(R))
M2=Q@diag(v)
```

## 1.4 Dynamic modeling

### 1.4.1 Principle

A robot (airplane, submarine, boat) can often be considered as a solid body whose inputs are the (tangential and angular) accelerations. These quantities are analytic functions of the forces that are at the origin of the robot's movement. For the dynamic modeling of a submarine the reference work is the book of Fossen [FOS 2002], but the related notions can also be used for other types of solid robots such as planes, boats or quadrotors. In order to obtain a dynamic model, it is sufficient to take the kinematic equations and to consider that the angular and tangential accelerations are caused by forces and torques which become the new inputs of our system. The link between the accelerations and the forces is done by Newton's second law (or the *fundamental principle of dynamics*). Thus, for instance if  $\mathbf{f}$  is the net force resulting from the external forces expressed in the inertial frame and  $m$  is the mass of the robot, we have

$$m\ddot{\mathbf{p}} = \mathbf{f}.$$

Since the speed and accelerations are generally measured by sensors embedded in the system, we generally prefer to express the speed and accelerations in the frame of the robot:

$$m\mathbf{a}_r = \mathbf{f}_r,$$

where  $\mathbf{f}_r$  is the applied forces vector expressed in the robot frame. The same type of relation, known as the *Euler's rotation equation*, exists for rotations. It is given by

$$\mathbf{I}\dot{\boldsymbol{\omega}}_r + \boldsymbol{\omega}_r \wedge (\mathbf{I}\boldsymbol{\omega}_r) = \boldsymbol{\tau}_r \quad (1.16)$$

where  $\boldsymbol{\tau}_r$  is the applied torque and  $\boldsymbol{\omega}_r$  is the rotation vector both expressed in the robot frame. The inertia matrix  $\mathbf{I}$  is attached to the robot (*i.e.*, computed in the robot frame) and we generally choose the robot frame to make  $\mathbf{I}$  diagonal. This relation is a consequence of the *Euler's second law* which states that in an inertial frame, the time derivative of the angular momentum  $\mathcal{L} = \mathbf{R} \cdot \mathbf{I} \cdot \mathbf{R}^T \cdot \boldsymbol{\omega}$  equals the applied torque  $\boldsymbol{\tau}$ . In the inertial frame, this can be expressed as

$$\begin{aligned} & \frac{d}{dt} \mathcal{L} = \boldsymbol{\tau} \\ \Leftrightarrow & \frac{d}{dt} (\mathbf{R} \cdot \mathbf{I} \cdot \boldsymbol{\omega}_r) = \mathbf{R} \cdot \boldsymbol{\tau}_r \\ \Leftrightarrow & \dot{\mathbf{R}} \mathbf{I} \cdot \boldsymbol{\omega}_r + \mathbf{R} \cdot \mathbf{I} \cdot \dot{\boldsymbol{\omega}}_r = \mathbf{R} \cdot \boldsymbol{\tau}_r \\ \Leftrightarrow & \mathbf{R}^T \dot{\mathbf{R}} \mathbf{I} \cdot \boldsymbol{\omega}_r + \mathbf{I} \cdot \dot{\boldsymbol{\omega}}_r = \boldsymbol{\tau}_r \\ \Leftrightarrow & \boldsymbol{\omega}_r \wedge (\mathbf{I}\boldsymbol{\omega}_r) + \mathbf{I}\dot{\boldsymbol{\omega}}_r = \boldsymbol{\tau}_r. \end{aligned}$$

### 1.4.2 Modeling a quadrotor

As an illustration, we consider a quadrotor (see Figure 1.8) for which we want a dynamic model. The robot has four propellers that can be tuned independently. This will allow us to control the attitude and position of the robot by changing the speeds of the motors.

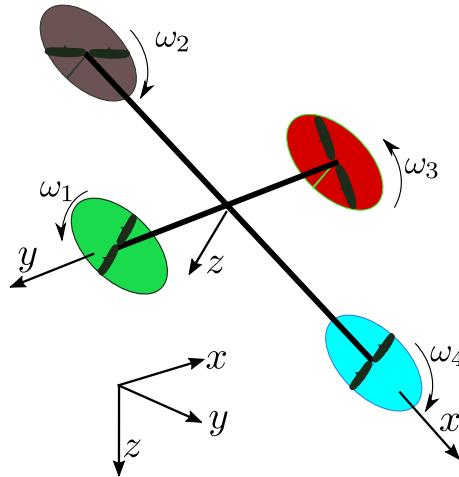


Figure 1.8: Quadrotor

We distinguish the front/rear propellers (blue and black), rotating clockwise and right/left propellers (red and green), rotating counterclockwise. The value of the force generated by the  $i$ th propeller is proportional to the squared speeds of the rotors, *i.e.*, is equal to  $\beta \cdot \omega_i \cdot |\omega_i|$ , where  $\beta$  is the thrust factor. Denote by  $\delta$  the drag factor and  $\ell$  the distance between any rotor and the center

of the robot. The forces and torques generated on the robot are

$$\begin{pmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} \beta & \beta & \beta & \beta \\ -\beta\ell & 0 & \beta\ell & 0 \\ 0 & -\beta\ell & 0 & \beta\ell \\ -\delta & \delta & -\delta & \delta \end{pmatrix} \cdot \begin{pmatrix} \omega_1 \cdot |\omega_1| \\ \omega_2 \cdot |\omega_2| \\ \omega_3 \cdot |\omega_3| \\ \omega_4 \cdot |\omega_4| \end{pmatrix}$$

where  $\tau_0$  is the total thrust generated by the rotors and  $\tau_1, \tau_2, \tau_3$  are the torques generated by differences in the rotor speeds.

We will now show that the state equations for the quadrotor are given by

$$\left\{ \begin{array}{lcl} \dot{\mathbf{p}} & = & \mathbf{R}(\varphi, \theta, \psi) \cdot \mathbf{v}_r \\ \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} & = & \begin{pmatrix} 1 & \tan \theta \sin \varphi & \tan \theta \cos \varphi \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \cdot \boldsymbol{\omega}_r \\ \dot{\mathbf{v}}_r & = & \mathbf{R}^T(\varphi, \theta, \psi) \cdot \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\frac{\tau_0}{m} \end{pmatrix} - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \dot{\boldsymbol{\omega}}_r & = & \mathbf{I}^{-1} \cdot \left( \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} - \boldsymbol{\omega}_r \wedge (\mathbf{I} \cdot \boldsymbol{\omega}_r) \right) \end{array} \right. \quad \begin{array}{l} (i) \\ (ii) \\ (iii) \\ (iv) \end{array}$$

where  $\mathbf{p}$  is the position and  $(\varphi, \theta, \psi)$ , are the Euler angles of the robot. The three first equations (i), (ii), (iii) correspond to the kinematic equations and have already been derived (see Equation (1.13)). Note that in Equation (iii), the acceleration  $\mathbf{a}_r$  (expressed in the robot frame) comes from the Newton's second law where the forces are the total force  $\tau_0$  and the weight  $mg$ :

$$m\mathbf{a}_r = \mathbf{R}^T(\varphi, \theta, \psi) \cdot \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\tau_0 \end{pmatrix}.$$

Since the gravity vector is expressed naturally in the inertial frame (contrary to the global force  $\tau_0$ ), we have to multiply the gravity vector by the Euler rotation matrix  $\mathbf{R}^T(\varphi, \theta, \psi)$ . Equation (iv) comes from the Euler's rotation equation (1.16)

$$\dot{\boldsymbol{\omega}}_r = \mathbf{I}^{-1} \cdot (\boldsymbol{\tau}_r - \boldsymbol{\omega}_r \wedge (\mathbf{I} \cdot \boldsymbol{\omega}_r))$$

where  $\boldsymbol{\tau}_r = (\tau_1, \tau_2, \tau_3)$  is the torque vector.



# Exercises

EXERCISE 1.1.– *Properties of the matrix  $\omega \wedge$*

See the correction video at <https://youtu.be/D3DbfurFWXo>

Let us consider the vector  $\omega = (\omega_x, \omega_y, \omega_z)$  and its associated skew symmetric matrix  $\wedge(\omega)$ .

- 1) Show that the eigenvalues of  $\wedge(\omega)$  are  $\{0, \|\omega\| \cdot i, -\|\omega\| \cdot i\}$ . Give an eigenvector associated with 0. Discuss.
  - 2) Show that the vector  $\wedge(\omega) \cdot \mathbf{x} = \omega \wedge \mathbf{x}$  is a vector perpendicular to  $\omega$  and  $\mathbf{x}$ , such that the trihedron  $(\omega, \mathbf{x}, \omega \wedge \mathbf{x})$  is direct.
  - 3) Show that the norm of  $\omega \wedge \mathbf{x}$  is surface of the parallelogram  $\mathcal{A}$  mediated by  $\omega$  and  $\mathbf{x}$ .
- 

EXERCISE 1.2.– *Jacobi identity*

See the correction video at <https://youtu.be/gD04pW7iYBw>

The Jacobi identity is written as:

$$\mathbf{a} \wedge (\mathbf{b} \wedge \mathbf{c}) + \mathbf{c} \wedge (\mathbf{a} \wedge \mathbf{b}) + \mathbf{b} \wedge (\mathbf{c} \wedge \mathbf{a}) = \mathbf{0}.$$

- 1) Show that this identity is equivalent to:

$$\wedge(\mathbf{a} \wedge \mathbf{b}) = \wedge(\mathbf{a}) \cdot \wedge(\mathbf{b}) - \wedge(\mathbf{b}) \cdot \wedge(\mathbf{a})$$

where  $\wedge(\omega)$  is the skew symmetric matrix associated to the vector  $\omega \in \mathbb{R}^3$ .

- 2) In the space of skew-symmetric matrices, the Lie bracket is defined as follows:

$$[\mathbf{A}, \mathbf{B}] = \mathbf{A} \cdot \mathbf{B} - \mathbf{B} \cdot \mathbf{A}.$$

Show that :

$$\wedge(\mathbf{a} \wedge \mathbf{b}) = [\wedge(\mathbf{a}), \wedge(\mathbf{b})].$$

- 3) In the Exercise 1.4, we will show that an infinitesimal rotation following the skew symmetric matrix  $\mathbf{A}$  is  $\exp(\mathbf{A} \cdot dt) \simeq \mathbf{I} + \mathbf{A} \cdot dt + \frac{\mathbf{A}^2}{2} \cdot dt^2 + o(dt^2)$ . Consider two skew-symmetric matrices  $\mathbf{A}, \mathbf{B}$  and define the two infinitesimal rotation matrices

$$\mathbf{R}_a = e^{\mathbf{A} \cdot dt} \text{ and } \mathbf{R}_b = e^{\mathbf{B} \cdot dt}.$$

Compute a Taylor expansion of second order associated to the rotation

$$\mathbf{R}_a^{-1}\mathbf{R}_b^{-1}\mathbf{R}_a\mathbf{R}_b.$$

From this result, give an interpretation of the Lie bracket  $[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA}$ .

- 4) Verify that the set  $(\mathbb{R}^3, +, \wedge, \cdot)$  forms a *Lie algebra*.
- 

EXERCISE 1.3.– *Varignon's formula*

See the correction video at [https://youtu.be/dMhwZLTA\\_cA](https://youtu.be/dMhwZLTA_cA)

Let us consider a solid body whose center of gravity remains at the origin of a Galilean coordinate system and is rotating around an axis  $\Delta$  with a rotation vector of  $\boldsymbol{\omega}$ . Give the equation of the trajectory of a point  $\mathbf{x}$  of the body.

---

EXERCISE 1.4.– *Rodrigues' formula*

See the correction video at <https://youtu.be/ySHG6Cvd-R8>

Let us consider a solid body whose center of gravity remains at the origin of a Galilean coordinate system and is rotating around an axis  $\Delta$ . The position of a point  $\mathbf{x}$  of the body satisfies the state equation (Varignon's formula):

$$\dot{\mathbf{x}} = \boldsymbol{\omega} \wedge \mathbf{x}$$

where  $\boldsymbol{\omega}$  is parallel to the axis of rotation  $\Delta$  and  $\|\boldsymbol{\omega}\|$  is the rotation speed of the body (in  $\text{rad.s}^{-1}$ ).

- 1) Show that this state equation can be written in the form:

$$\dot{\mathbf{x}} = \mathbf{Ax}.$$

Explain why the matrix  $\mathbf{A}$  is often denoted by  $\boldsymbol{\omega} \wedge$ .

- 2) Give the expression of the solution of the state equation.

3) Deduce from this that the expression of the rotation matrix  $\mathbf{R}$  with angle  $\|\boldsymbol{\omega}\|$  around  $\boldsymbol{\omega}$  is given by the following formula, referred to as *Rodrigues' formula*:

$$\mathbf{R} = e^{\boldsymbol{\omega} \wedge}.$$

4) Calculate the eigenvalues of  $\mathbf{A}$  and show that  $\boldsymbol{\omega}$  is the eigenvector associated with the zero eigenvalue. Discuss.

- 5) What are the eigenvalues of  $\mathbf{R}$  ?

6) Using the previous questions, give the expression of a rotation around the vector  $\boldsymbol{\omega} = (1, 0, 0)$  of angle  $\alpha$ .

- 7) Write an expression of the Euler matrix that uses Rodrigues' formula.
-

EXERCISE 1.5.– *Quaternions*

See the correction video at [https://youtu.be/d5dRd\\_SjDTU](https://youtu.be/d5dRd_SjDTU)

Quaternion were discovered by W. R. Hamilton are also used to represent a rotation in a 3-dimensional space (see, e.g., [COR 11]). A quaternion  $\dot{q}$  is an extension of the complex number. It corresponds to a scalar  $s$  plus a vector  $\mathbf{v}$  of  $\mathbb{R}^3$ . We use the equivalent following notations:

$$\dot{q} = s + v_1 i + v_2 j + v_3 k = s + \langle v_1, v_2, v_3 \rangle = s + \langle \mathbf{v} \rangle,$$

where

$$i^2 = j^2 = k^2 = ijk = -1.$$

- 1) From these relations, fill the following multiplication table:

.	1	$i$	$j$	$k$
1	1	$i$	$j$	$k$
$i$	$i$	?	?	?
$j$	$j$	?	?	?
$k$	$k$	?	?	?

- 2) A unit-quaternion  $\dot{q}$  is a quaternion with a unit magnitude:

$$|\dot{q}|^2 = s^2 + v_1^2 + v_2^2 + v_3^2 = 1.$$

Consider the rotation obtained by a rotation around the unit vector  $\mathbf{v}$  with an angle  $\theta$ . From the Rodrigues formula, we know that the corresponding rotation matrix is  $\exp(\theta \cdot \mathbf{v} \wedge)$ . To this rotation, we associate the quaternion:

$$\dot{q} = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right) \cdot \langle \mathbf{v} \rangle.$$

Check that this quaternion has a unit magnitude.

- 3) Show that the quaternion  $\dot{q}$  and its opposite  $-\dot{q}$ , both correspond to the same rotation in  $\mathbb{R}^3$ .  
4) Assuming that the composition of rotations corresponds to a multiplication of quaternions, show that

$$(s + \langle \mathbf{v} \rangle)^{-1} = s + \langle -\mathbf{v} \rangle.$$

- 5) We consider the Euler rotation matrix with  $\psi = \theta = \varphi = \frac{\pi}{2}$ . It corresponds to a rotation around the unit vector  $\mathbf{v}$  with an angle  $\alpha$ . Give the values of  $\mathbf{v}$  and  $\alpha$  using 3 methods: the geometrical method (using hands only), the matrices and the quaternions.

EXERCISE 1.6.– *Car on the sphere*

See the correction video at <https://youtu.be/1pgs4DBQosk>

Consider a car moving in the plane described by the state equations:

$$\begin{cases} \dot{x}_1 = x_4 \cos x_3 \\ \dot{x}_2 = x_4 \sin x_3 \\ \dot{x}_3 = u_1 \\ \dot{x}_4 = u_2 \end{cases}$$

where  $(x_1, x_2, x_3)$  is the pose of the car,  $x_4$  is the speed,  $u_1$  is the rotation rate and  $u_2$  is the acceleration (see 1.9, left).

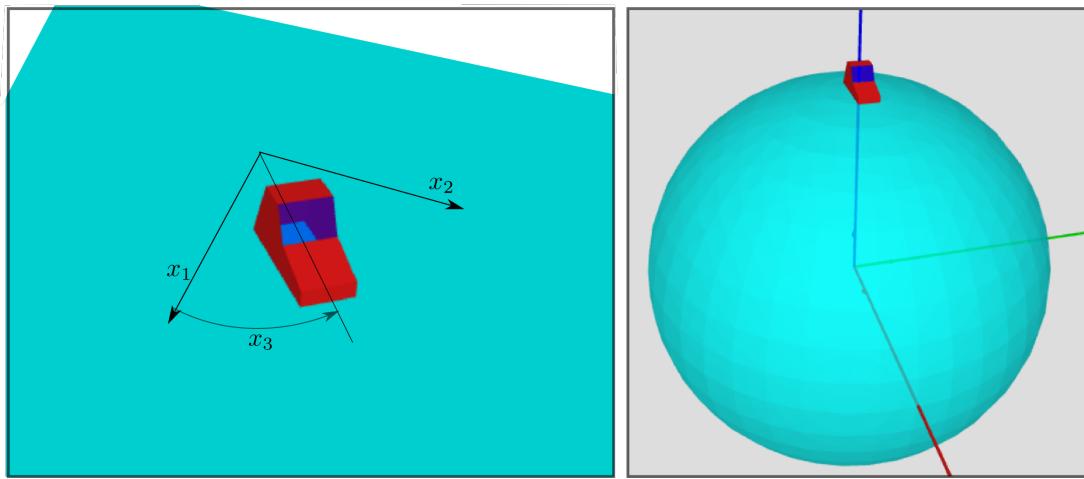


Figure 1.9: Left: car on the plane; Right: car on the sphere

For the input, we choose

$$\mathbf{u} = \begin{pmatrix} 0.1 \cdot \cos(0.1 \cdot t) + 0.02 \\ 1 - x_4 \end{pmatrix}$$

and for the initial state, we take  $\mathbf{x} = (0, 0, 0, 0)$ .

- 1) What is the shape of the manifold corresponding to the state space. Simulate the system for  $t \in [0, 200]$ .
- 2) We take the same car, with the same input  $\mathbf{u}$ . But now, the car moves on a sphere with diameter  $r = 20$  as illustrated by Figure 1.9, right. What is the shape of the state-space?
- 3) Provide a simulation of the car moving the sphere. The simulation should avoid the singularities.

#### EXERCISE 1.7.– Schuler pendulum

See the correction video at <https://youtu.be/15pV-tfFLV8>

One of the fundamental components of an inertial unit is the inclinometer. This sensor gives the vertical direction. Traditionally, we use a pendulum (or a plumb line) for this. However, when we are moving, due to the accelerations the pendulum starts to oscillate and it can no longer be used to measure the vertical direction. Here, we are interested in designing a pendulum for which any horizontal acceleration does not lead to oscillations. Let us consider a pendulum with two masses  $m$  at each end, situated at a distance  $\ell_1$  and  $\ell_2$  from the axis of rotation of the rod (see Figure 1.10). The axis moves over the surface of the Earth. We assume that  $\ell_1$  and  $\ell_2$  are small in comparison to Earth's radius  $r$ .

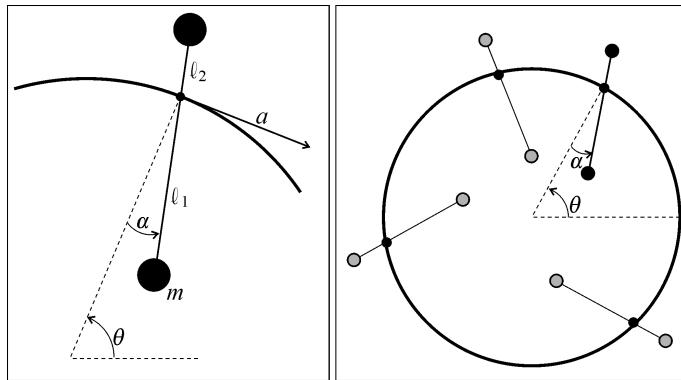


Figure 1.10: Inclinometer pendulum moving over the surface of the Earth

1) Find the state equations of the system.

2) Let us assume that  $\alpha = \dot{\alpha} = 0$ . For which values of  $\ell_1$  and  $\ell_2$  does the pendulum remain vertical, for any horizontal movement of the pendulum? What values does  $\ell_2$  have to take if we let  $\ell_1 = 1$  m and we take  $r = 6400$  km for the Earth's radius?

3) Let us assume that, as a result of disturbances, the pendulum starts to oscillate. The period of these oscillations is called the *Schuler period*. Calculate this period.

4) Simulate the system graphically by taking a Gaussian white noise as the acceleration input. Since the system is conservative, an Euler integration method will not perform well (the pendulum would gain energy). A higher-order integration scheme, such as that of *Runge Kutta*, should be used. This is given by:

$$\mathbf{x}(t+dt) \simeq \mathbf{x}(t) + dt \cdot \left( \frac{\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))}{4} + \frac{3}{4}\mathbf{f}(\mathbf{x}(t) + \frac{2dt}{3}\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{u}(t + \frac{2}{3}dt)) \right).$$

For an easier graphical representation, we will take  $r = 10$  m,  $\ell_1 = 1$  m and  $g = 10$  ms<sup>-2</sup>. Discuss the results.

#### EXERCISE 1.8.– Immersion

See the correction video at <https://youtu.be/E-XX52wbBQ8>

Consider the pendulum described by the state equations:

$$\left\{ \begin{pmatrix} \dot{\theta} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} \omega \\ -\sin \theta - \frac{1}{2}\omega \end{pmatrix} \right.$$

- 1) Considering that the two states  $(\theta, \omega)$  and  $(\theta + 2k\pi, \omega), k \in \mathbb{Z}$ , are identical, show that the state space corresponds to a cylinder?
  - 2) We want to represent the cylinder which is a two-dimensional manifold in the larger dimensional Cartesian space  $\mathbb{R}^3$ . We call this representation an *immersion*. Set  $x_1 = \cos \theta, x_2 = \sin \theta, x_3 = \omega$ . Give the state equations in the  $\mathbf{x}$ -space. Which constraint should satisfy  $\mathbf{x}$ ?
  - 3) Provide a Runge-Kutta simulation on the  $\mathbf{x}$ -space of the system. The initial vector is chosen as  $(-1, 0, 6)$ . Draw the corresponding trajectory and interpret.
- 

#### EXERCISE 1.9.– Car on the torus

See the correction video at <https://youtu.be/TcZ0f1iVB-A>

Consider a car moving on the torus, parameterized by the two radius  $r_1, r_2$ , as illustrated by Figure 1.11. The pose of the car is represented by 3 angles  $x_1, x_2, x_3$ . The speed of the car is always equal to 1. The car has a single input  $u$  corresponding to its rotation rate with respect to the ground.

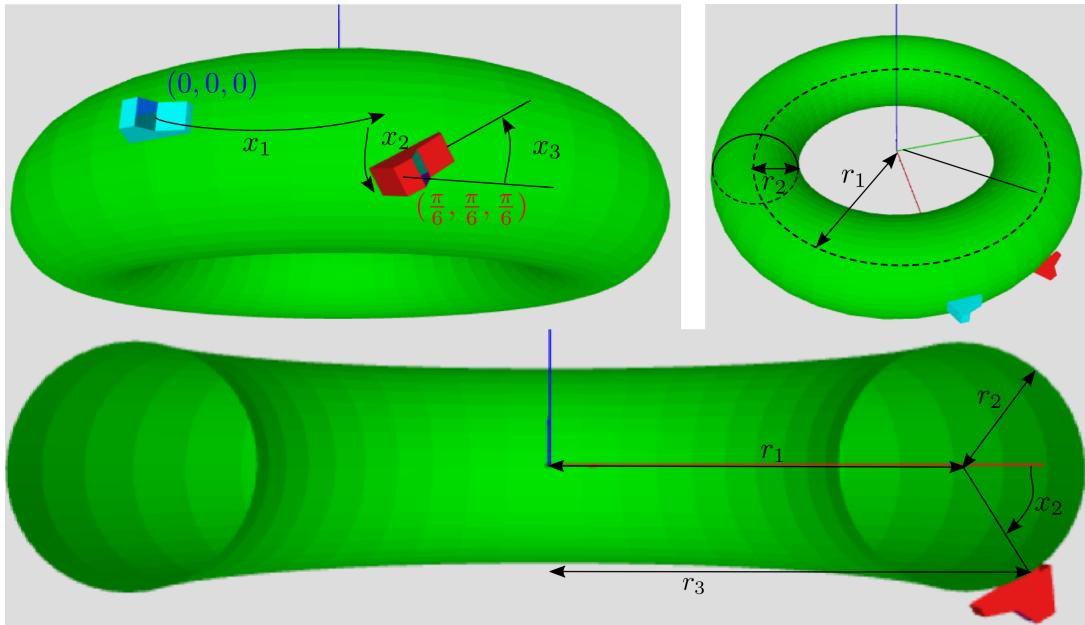


Figure 1.11: Representation of the pose of the car on the torus

- 1) What is the shape of the manifold corresponding to the state space. Give the state equations of the system.
- 2) Simulate the system for  $r_1 = 10, r_2 = 6$  and  $t \in [0, 100]$ . The initial state is taken as  $x_1 = x_2 = x_3 = 0$ . For the control, we choose  $u = 0.1$ .

EXERCISE 1.10.– *Brake detector*

See the correction video at <https://youtu.be/wJnWXbDz38g>

We will now look at a problem that involves basis changes and rotation matrices. A car is preceded by another car **m** (which we will assume to be a point). We attach to this car the coordinate system  $\mathcal{R}_1 : (\mathbf{o}_1, \mathbf{i}_1, \mathbf{j}_1)$  as represented on Figure 1.12. The coordinate system  $\mathcal{R}_0 : (\mathbf{o}_0, \mathbf{i}_0, \mathbf{j}_0)$  is a ground frame assumed to be fixed.

This car is equipped with the following sensors:

- several odometers placed on the rear wheels allowing to measure the speed  $v$  of the center of the rear axle ;
- a gyro giving the angular speed of the car  $\dot{\theta}$ , as well as the angular acceleration  $\ddot{\theta}$  ;
- an accelerometer placed at  $\mathbf{o}_1$  allowing to measure the acceleration vector  $(\alpha, \beta)$  of  $\mathbf{o}_1$  expressed in the coordinate system  $\mathcal{R}_1$  ;
- using two radars placed at the front, our car is capable of (indirectly) measuring the coordinates  $(a_1, b_1)$  of the point **m** in the coordinate system  $\mathcal{R}_1$  as well as the first two derivatives  $(\dot{a}_1, \dot{b}_1)$  and  $(\ddot{a}_1, \ddot{b}_1)$ .

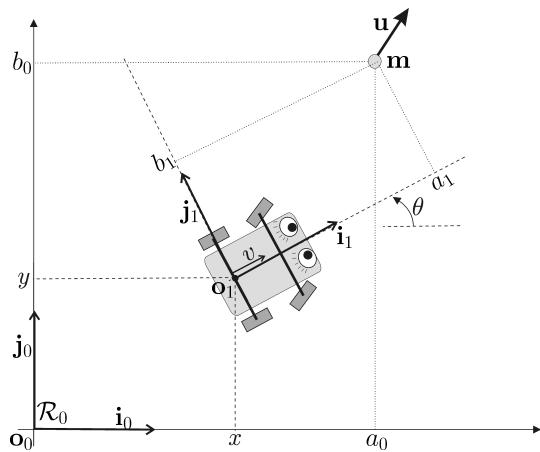


Figure 1.12: Car trying to detect whether the point **m** is braking

On the other hand, the car is not equipped with a positioning system (such as a GNSS, for example) that would allow it to gain knowledge of  $x, y, \dot{x}, \dot{y}$ . It does not have a compass for measuring the angle  $\theta$ . The quantities in play are the following:

Measured  $v, \dot{\theta}, a_1, b_1, \alpha, \beta$ .  
Unknown  $x, y, \dot{x}, \dot{y}, \theta, a_0, b_0$

When a quantity is measured, we assume that its differentials are also measured, but not their primitive. For example  $\dot{a}_1, \dot{b}_1, \ddot{a}_1, \ddot{b}_1$  are considered to be measured since  $a_1, b_1$  are measured. However,  $\dot{\theta}$  is measured but  $\theta$  isn't. We do not know the state equations of our car. The goal of this problem is finding a condition on the measured variables (and their derivatives) that will allow us to tell whether the point  $\mathbf{m}$  is braking or not. We understand that such a condition would allow us to build a warner informing us that the preceding vehicle is braking, even when its rear brake lights are not visible (fog, trailer without brake lights) or defective.

- 1) By expressing the Chasle relation ( $\mathbf{o}_0\mathbf{m} = \mathbf{o}_0\mathbf{o}_1 + \mathbf{o}_1\mathbf{m}$ ) in the coordinate system  $\mathcal{R}_0$ , show the basis change formula:

$$\begin{pmatrix} a_0 \\ b_0 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{R}_\theta \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$$

where  $\mathbf{R}_\theta$  is a rotation matrix.

- 2) Show that  $\mathbf{R}_\theta^T \dot{\mathbf{R}}_\theta$  is a skew-symmetric matrix and find its expression.
- 3) Let  $\mathbf{u}$  be the speed vector of the point  $\mathbf{m}$  viewed by a fixed observer. Find an expression  $\mathbf{u}_{|\mathcal{R}_1}$  of the speed vector  $\mathbf{u}$  expressed in the coordinate system  $\mathcal{R}_1$ . Express  $\mathbf{u}_{|\mathcal{R}_1}$  in function of the measured variables  $v, \dot{\theta}, a_1, \dot{a}_1, b_1, \dot{b}_1$ .
- 4) We will now use the accelerometer of our vehicle, which gives us the acceleration vector  $(\alpha, \beta)$  of  $\mathbf{o}_1$ , expressed in  $\mathcal{R}_1$ . By differentiating  $\mathbf{m}_{|\mathcal{R}_0}$  two times, give the expression of the acceleration  $\mathbf{a}_{|\mathcal{R}_0}$  of  $\mathbf{m}$  in the coordinate system  $\mathcal{R}_0$ . Deduce from this its expression  $\mathbf{a}_{|\mathcal{R}_1}$  in the coordinate system  $\mathcal{R}_1$ . Give the expression of  $\mathbf{a}_{|\mathcal{R}_1}$  only in function of the measured variables.
- 5) Find a condition on the measurements  $(v, \dot{\theta}, a_1, b_1, \dot{a}_1, \dot{b}_1, \ddot{a}_1, \ddot{b}_1, \alpha, \beta)$  which allows to detect whether the vehicle in front is braking.

#### EXERCISE 1.11.– Manipulator robot

See the correction video at <http://youtu.be/TPlHx0PE6P4>

A manipulator robot, such as *Staubli* represented on Figure 1.13, is composed of several rigid arms. We retrieve the coordinates of the end effector, at the extremity of the robot, using a series of geometric transformations. We can show that a parametrization with four degrees of freedom allows to represent these transformations. There are several possible parametrizations, each with its own advantages and disadvantages. The most widely used one is probably the *Denavit-Hartenberg* parametrization. In the case where the articulations are rotational joints (as is the case of the Staubli robot where the joins can turn), the parametrization represented by the figure might prove to be practical since it makes drawing the robot easier. This transformation is the composition of four elementary transformations: (i) a translation of length  $r$  following  $z$ ; (ii) a translation of length  $d$  following  $x$ ; (iii) a rotation of  $\alpha$  around  $y$  and (iv) a rotation of  $\theta$  (the variable activated around  $z$ ). Using the figure for drawing the arms and the photo for the robot, perform a realistic simulation of the robot's movement.

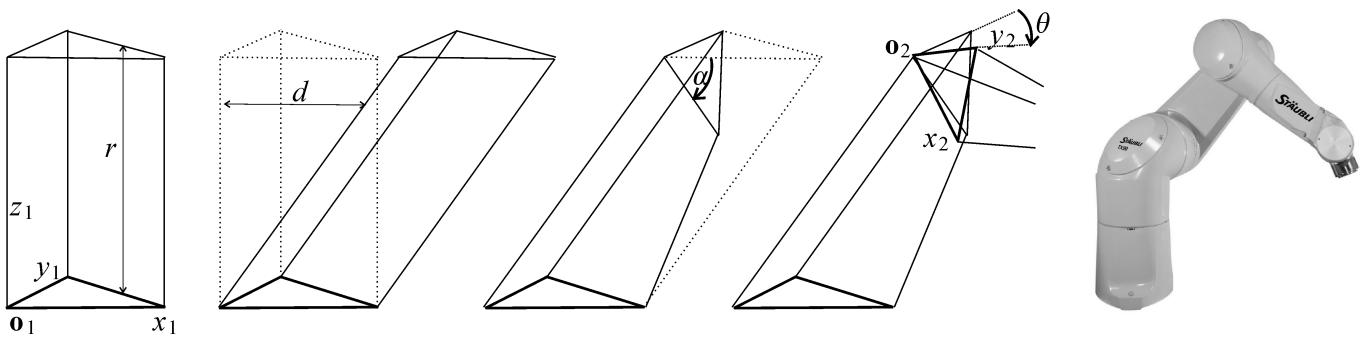


Figure 1.13: Parametrization for the direct geometric model of a manipulator robot

### EXERCISE 1.12.– Modeling an underwater robot

See the correction video at <https://youtu.be/Ox8zavW9WYw>

The robot we will be modeling is the Redermor (*greyhound of the sea* in the Breton language). It is represented on Figure 1.14. It is an entirely autonomous underwater robot. This robot, developed by GESMA (Groupe d'Etude Sous-Marine de l'Atlantique - *Atlantic underwater research group*), has a length of 6 m, a diameter of 1 m and a weight of 3 800 kg. It has an efficient propulsion and control system with the aim of finding mines on the seabed.



Figure 1.14: *Redermor* built by GESMA (Groupe d'Etude Sous-Marine de l'Atlantique - *Atlantic underwater research group*), on the water surface, still close to the boat it was launched from

Let us build a local coordinate system  $\mathcal{R}_0 : (\mathbf{o}_0, \mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0)$  over the area traveled by the robot. The point  $\mathbf{o}_0$  is placed on the surface of the ocean. The vector  $\mathbf{i}_0$  indicates north,  $\mathbf{j}_0$  indicates east and  $\mathbf{k}_0$  is oriented towards the center of the Earth. Let  $\mathbf{p} = (p_x, p_y, p_z)$  be the coordinates of the center of the robot expressed in the coordinate system  $\mathcal{R}_0$ . The state variables of the underwater robot are its position  $\mathbf{p}$  in the coordinate system  $\mathcal{R}_0$ , its tangential  $v$  and its three Euler angles  $\varphi, \theta, \psi$ . Its

inputs are the tangential acceleration  $\dot{v}$  as well as three control rudders which act respectively on  $\omega_x, \omega_y, \omega_z$ . More formally, we have:

$$\begin{cases} u_1 = \dot{v} \\ vu_2 = \omega_y \\ vu_3 = \omega_z \\ u_4 = \omega_x \end{cases}$$

where the factor  $v$  preceding  $u_2, u_3$  indicates that the robot is only able to turn left/right (through  $u_3$ ) or up/down (through  $u_2$ ) when it is advancing. Give the kinematic state model for this system.

---

#### EXERCISE 1.13.– 3D robot graphics

See the correction video at <https://youtu.be/t4ay2KENg-k>

Drawing two- or three-dimensional robots or objects on the screen is widely used for simulation in robotics. The classic method (used by OPENGL) relies on modeling the posture of objects using a series of affine transformations (rotations, translations, homotheties) of the form:

$$\begin{aligned} \mathbf{f}_i : \mathbb{R}^n &\rightarrow \mathbb{R}^n \\ \mathbf{x} &\mapsto \mathbf{A}_i \mathbf{x} + \mathbf{b}_i \end{aligned}$$

with  $n = 2$  or  $3$ . However, the manipulation of compositions of affine functions is less simple than that of linear applications. The idea of the transformation in *homogeneous coordinates* is to transform a system of affine equations into a system of linear equations. Notice first of all that an affine equation of the type  $\mathbf{y} = \mathbf{Ax} + \mathbf{b}$  can be written as:

$$\begin{pmatrix} \mathbf{y} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}.$$

We will thus define the *homogeneous transformation* of a vector as follows:

$$\mathbf{x} \mapsto \mathbf{x}_h = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}.$$

Thus, an equation of the type:

$$\mathbf{y} = \mathbf{A}_3 (\mathbf{A}_2 (\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3$$

involving the composition of three affine transformations, can be written as:

$$\mathbf{y}_h = \begin{pmatrix} \mathbf{A}_3 & \mathbf{b}_3 \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{A}_1 & \mathbf{b}_1 \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{x}_h.$$

A *sketch* is a matrix with two or three rows (depending on the object being in the plane or in space) and  $n$  columns representing the  $n$  vertices of a rigid polygon embodying the object. It is important

that the union of all the segments formed by two consecutive points of the sketch forms all the vertices of the polygon that we wish to represent.

1) Let us consider the underwater robot (or AUV for *Autonomous Underwater Vehicle*) whose sketch matrix in homogeneous coordinates is :

$$\mathbf{M} = \begin{pmatrix} 0 & 0 & 10 & 0 & 0 & 10 & 0 & 0 \\ -1 & 1 & 0 & -1 & -0.2 & 0 & 0.2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Draw this sketch in perspective view on a piece of paper.

2) The state equations of the robot are the following:

$$\begin{cases} \dot{p}_x = v \cos \theta \cos \psi \\ \dot{p}_y = v \cos \theta \sin \psi \\ \dot{p}_z = -v \sin \theta \\ \dot{v} = u_1 \\ \dot{\varphi} = -0.1 \sin \varphi \cdot \cos \theta + \tan \theta \cdot v \cdot (\sin \varphi \cdot u_2 + \cos \varphi \cdot u_3) \\ \dot{\theta} = \cos \varphi \cdot v \cdot u_2 - \sin \varphi \cdot v \cdot u_3 \\ \dot{\psi} = \frac{\sin \varphi}{\cos \theta} \cdot v \cdot u_2 + \frac{\cos \varphi}{\cos \theta} \cdot v \cdot u_3 \end{cases}$$

where  $(\varphi, \theta, \psi)$  are the three Euler angles. The inputs of the system are the tangential acceleration  $u_1$ , the pitch  $u_2$  and the yaw  $u_3$ . The state vector is therefore equal to  $\mathbf{x} = (p_x, p_y, p_z, v, \varphi, \theta, \psi)$ . Give a function capable of drawing the robot in 3D together with its shadow, in the plane  $x-y$ . Verify that the drawing is correct by moving the six degrees of freedom of the robot one by one. Obtain a three-dimensional representation such as the one illustrated in Figure 1.15.

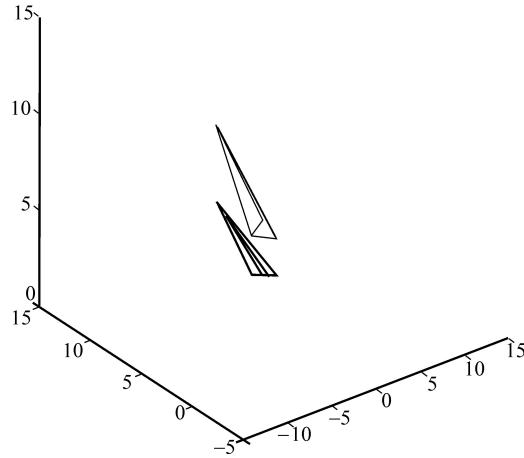


Figure 1.15: 3D representation of the robot together with its shadow in the horizontal plane

3) By using the relation:

$$\omega|_{\mathcal{R}_0} = \begin{pmatrix} \cos \theta \cos \psi & -\sin \psi & 0 \\ \cos \theta \sin \psi & \cos \psi & 0 \\ -\sin \theta & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

draw the instantaneous rotation vector of the robot.

- 4) Simulate the robot in various conditions using an Euler method.

#### EXERCISE 1.14.– *Dzhanibekov effect*

See the correction video at <https://youtu.be/EA1Rh5MgGKI>

The Dzhanibekov effect corresponds to the instability of the rotation of a rigid body object around its second principal axes. We consider a parallelepiped of mass  $m$  given in Figure 1.16. We assume that the density is uniform.

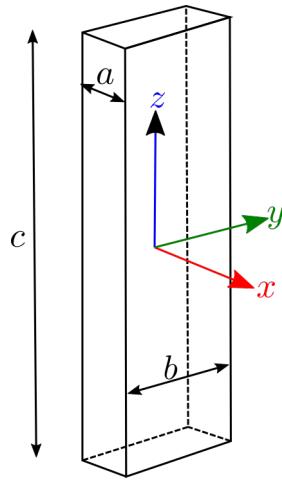


Figure 1.16: Parallelepiped body used to illustrate the Dzhanibekov effect

- 1) Recall that the inertia matrix of a solid body occupying the volume  $V$  is

$$\mathbf{I} = \int_V \rho(x, y, z) \begin{pmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{pmatrix} dx dy dz$$

where  $\rho(x, y, z)$  is the density. Compute the inertia matrix of the parallelepiped.

- 2) The Euler equation of a free rigid body is given by

$$\dot{\boldsymbol{\omega}}_r = -\mathbf{I}^{-1} \cdot (\boldsymbol{\omega}_r \wedge (\mathbf{I} \cdot \boldsymbol{\omega}_r))$$

where  $\boldsymbol{\omega}_r = (\omega_1, \omega_2, \omega_3)$  is the rotation vector expressed in the body frame. Give the state equation governing the rotation of our parallelepiped. The state vector is taken as  $(\omega_1, \omega_2, \omega_3)$ .

- 3) Give the equilibrium points of the system in the state-space. Conclude about the rotation of any solid asteroid in the rotating space.

- 4) Study the stability of the rotation along the steady states.

- 5) Take  $a = 0.4, b = 1, c = 3, m = 1$ . Illustrate by a simulation, an unstable situation with a rotation of the parallelepiped rotating along one of its principle axis.

**EXERCISE 1.15.– Euler field**

See the correction video at <https://youtu.be/in5wrQkVv4A>

The goal of this exercise is to provide a geometrical interpretation of the behavior of a solid body in rotation. We consider the Euler equation given by

$$\dot{\boldsymbol{\omega}}_r = -\mathbf{I}^{-1} \cdot (\boldsymbol{\omega}_r \wedge (\mathbf{I} \cdot \boldsymbol{\omega}_r))$$

where  $\boldsymbol{\omega}_r$  is the rotation vector expressed in the body frame and  $\mathbf{I}$  is the inertia matrix given by

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

- 1) Show that the kinetic energy given by

$$E_K = \frac{1}{2} \boldsymbol{\omega}_r^T \mathbf{I} \boldsymbol{\omega}_r$$

is constant.

2) Draw in three dimensional figure, the ellipsoid  $\mathcal{E}$  corresponding to the set of all  $\boldsymbol{\omega}_r$  such that energy equal to  $E_0 = 10J$ . Draw the vector field describing the evolution of  $\boldsymbol{\omega}_r$  on this ellipsoid.

3) Simulate with a Runge-Kutta method the evolution of the rotation vector for an initial condition  $\boldsymbol{\omega}_r(0) = (4, \sqrt{2}, 0)$ . Draw the corresponding trajectory on the ellipsoid  $\mathcal{E}$ . Interpret the result.

- 4) Simulate an evolution for  $\boldsymbol{\omega}_r(0) = (0.01, \sqrt{10}, 0)$ . Conclude.

**EXERCISE 1.16.– Floating wheel**

See the correction video at <https://youtu.be/mTh7n6mg03g>

Consider a wheel floating and spinning in the space as shown on Figure 1.17.

We assume that the wheel is solid, similar to a homogeneous disk of mass  $m = 1$  and radius  $\rho = 1$ . Its inertia matrix is given by:

$$\mathbf{I} = \begin{pmatrix} \frac{m\rho^2}{2} & 0 & 0 \\ 0 & \frac{m\rho^2}{4} & 0 \\ 0 & 0 & \frac{m\rho^2}{4} \end{pmatrix}$$

- 1) Give the state equations of the wheel. The state variables we choose are (i) the coordinates  $\mathbf{p} = (x, y, z)$  of the center of the wheel, (ii) the orientation  $(\varphi, \theta, \psi)$  of the wheel, (iii) the speed  $\mathbf{v}_r$  of

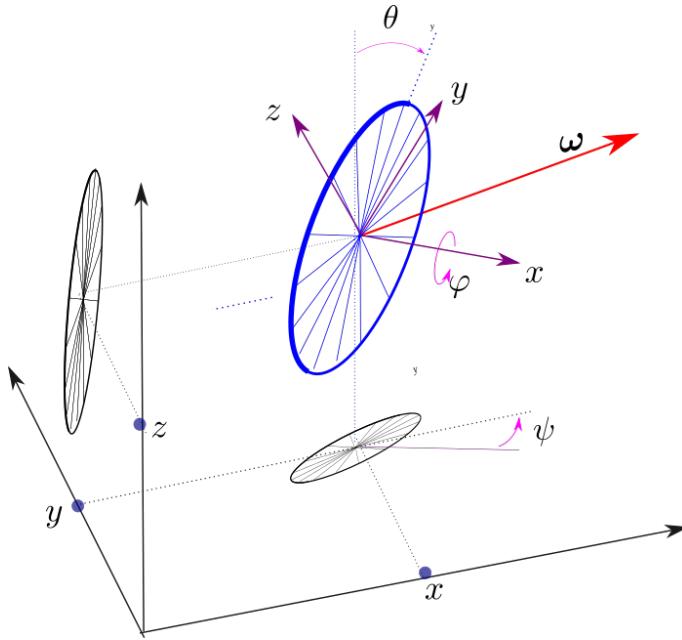


Figure 1.17: Floating wheel

the center of the wheel center in the wheel frame, and (iv) the rotation vector  $\omega_r$  expressed in the wheel frame. We thus have a system of order 12.

2) The torque-free precession occurs when the angular velocity vector changes its orientation with time. Provide a simulation illustrating the precession.

3) Show that the angular momentum  $\mathcal{L} = \mathbf{R} \cdot \mathbf{I} \cdot \mathbf{R}^T \cdot \boldsymbol{\omega}$  and the kinetic energy  $E_K = \frac{1}{2}\mathcal{L}^T \cdot \boldsymbol{\omega}$  are both constant. Verify with a simulation.

4) Using symbolic computation, compute the Jacobian matrix of the evolution function  $\mathbf{f}$  at point

$$\mathbf{x}_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, \omega_x, \omega_y, \omega_z)^T.$$

Give an interpretation.

#### EXERCISE 1.17.– Schuler oscillations in an inertial unit

See the correction video at <https://youtu.be/9vsZCnPXCPo>

Consider the Earth which is considered as static, without rotation, with a gravity field directed toward the center and with a magnitude  $g = 9.81\text{ms}^{-2}$ . An inertial unit is fixed in a robot  $\mathcal{R}_1$  at the surface of the Earth of radius  $r$ . Its position is with  $\mathbf{p} = (r, 0, 0)$  and its Euler angles rate is  $(\varphi, \theta, \psi) = (0, 0, 0)$  as in Figure 1.18. The robot  $\mathcal{R}_1$  will always be static in the exercise.

1) Give the values for  $\mathbf{a}_r^{mes}$  and  $\boldsymbol{\omega}_r$  collected by the inertial unit of  $\mathcal{R}_1$ .

2) We now want to lure the inertial unit by initializing the robot at some other place, say  $\mathcal{R}_2$ , and finding a trajectory so that the unit believes that it is fixed in  $\mathcal{R}_1$ . More precisely, we want to find a motion  $\mathbf{p}(t) = (x(t), y(t), z(t))$  for  $\mathcal{R}_2$  so that the inertial unit senses exactly the inputs  $\mathbf{a}_r^{mes}, \boldsymbol{\omega}_r$  of

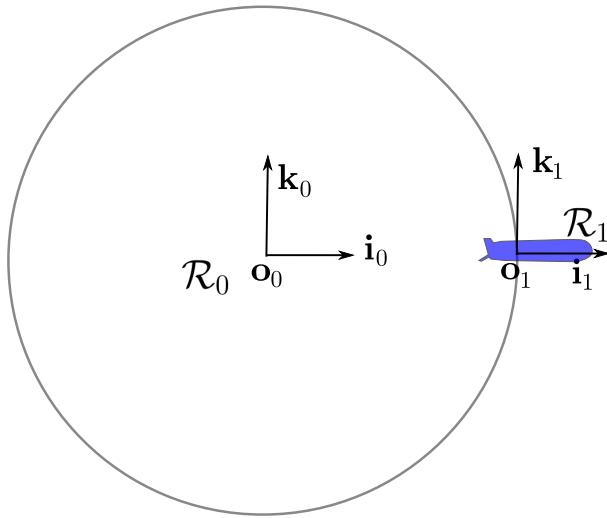


Figure 1.18: The inertial unit inside the robot is static

$\mathcal{R}_1$ . For this, we assume that at time  $t = 0$ ,  $(\varphi, \theta, \psi) = (0, 0, 0)$  for  $\mathcal{R}_2$ . For simplicity, we also assume that  $\mathcal{R}_2$  remains in the plane  $y = 0$ . Simplify the state equations for  $\mathcal{R}_2$  in our specific case.

3) Provide a simulation of the trajectory of  $\mathcal{R}_2$ , initialized at  $\mathbf{p}(0) = (r, 0, 1)^T$ . Interpret.

4) Linearize the state equation for  $\mathcal{R}_2$  at the state vector corresponding to  $\mathcal{R}_1$ . Compute the eigen values and give an interpretation.

5) Assuming that the inertial unit is not perfectly initialized, what can you conclude for the corresponding error propagation.

#### EXERCISE 1.18.– *Lie bracket for control*

See the correction video at <https://youtu.be/PfitQ7ZuNZs>

The motivation of Lie bracket for control is illustrated by Figure 1.19 on the car parking problem. We would like to move the blue car sideway between the two red cars already parked. Moving sideway is not possible, but we can approach this direction using many small forward-backward maneuvers. Building this new sideway direction corresponds to the Lie bracket operator between two dynamics. The goal of this exercise is two illustrate Lie brackets on a simple Dubins car.

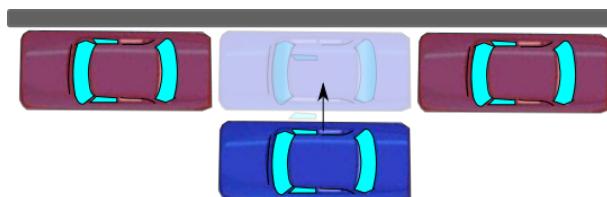


Figure 1.19: The inertial unit inside the robot is static

Consider two vector fields  $\mathbf{f}$  and  $\mathbf{g}$  of  $\mathbb{R}^n$  corresponding to two state equations  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$  and  $\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x})$ .

We define the Lie bracket between these two vector fields as

$$[\mathbf{f}, \mathbf{g}] = \frac{d\mathbf{g}}{d\mathbf{x}} \cdot \mathbf{f} - \frac{d\mathbf{f}}{d\mathbf{x}} \cdot \mathbf{g}.$$

We can check that the set of vector fields equipped with the Lie bracket is a Lie algebra. The proof is not difficult but tedious and will not be asked in this exercise.

- 1) Consider the linear vector fields  $\mathbf{f}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ ,  $\mathbf{g}(\mathbf{x}) = \mathbf{B} \cdot \mathbf{x}$ . Compute  $[\mathbf{f}, \mathbf{g}](\mathbf{x})$ .
- 2) Consider the system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \cdot u_1 + \mathbf{g}(\mathbf{x}) \cdot u_2.$$

We propose to apply the following cyclic sequence:

$$\begin{array}{lllll} t \in [0, \delta] & t \in [\delta, 2\delta] & t \in [2\delta, 3\delta] & t \in [3\delta, 4\delta] & t \in [4\delta, 5\delta] \dots \\ \mathbf{u} = (1, 0) & \mathbf{u} = (0, 1) & \mathbf{u} = (-1, 0) & \mathbf{u} = (0, -1) & \mathbf{u} = (1, 0) \dots \end{array}$$

where  $\delta$  is an infinitesimal time period  $\delta$ . This periodic sequence will be denoted by

$$\{(1, 0), (0, 1), (-1, 0), (0, -1)\}$$

Show that

$$\mathbf{x}(t + 2\delta) = \mathbf{x}(t - 2\delta) + [\mathbf{f}, \mathbf{g}](\mathbf{x}(t)) \delta^2 + o(\delta^2).$$

3) Which periodic sequence should we take in order to follow the field  $\nu \cdot [\mathbf{f}, \mathbf{g}]$ . We will first consider the case  $\nu \geq 0$  and then the case  $\nu \leq 0$ .

- 4) Propose a controller such that the system becomes

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \cdot a_1 + \mathbf{g}(\mathbf{x}) \cdot a_2 + [\mathbf{f}, \mathbf{g}](\mathbf{x}) \cdot a_3$$

where  $\mathbf{a} = (a_1, a_2, a_3)$  is the new input vector.

- 5) Consider a Dubins car described by the following state equations:

$$\begin{cases} \dot{x} = u_1 \cos \theta \\ \dot{y} = u_1 \sin \theta \\ \dot{\theta} = u_2 \end{cases}$$

where  $u_1$  is the speed of the cart,  $\theta$  its orientation and  $(x, y)$  the coordinates of its center. Using Lie Brackets, add a new input to the system, *i.e.*, a new direction of control.

6) Propose a simulation to check the good behavior of your controller. To do this take  $\delta$  small enough to be consistent with the second order Taylor approximation but large with respect to the sampling time  $dt$ . For instance, we may take  $\delta = \sqrt{dt}$ . The initial state vector is taken as  $\mathbf{x}(0) = (0, 0, 1)$ .

7) Propose a controller so that the car goes up. Consider the same questions, where the car goes down, right and left, respectively.

## EXERCISE 1.19.– Follow the equator

See the correction video at <https://youtu.be/8brHzqFHHsG>

We consider a mobile body moving around a planet, the Earth, with a radius  $\rho_E$  rotating along the vertical axis with a rotation rate  $\dot{\psi}_E$ .

1) Draw the Earth rotating with a static body at position  $\mathbf{p}$  with the following frames (see Figure 1.20):

1.  $\mathcal{R}_0$  is the Earth-centered inertial frame, fixed with respect to the stars
2.  $\mathcal{R}_1$  is the Earth frame and turning with the Earth with a rotational speed  $\dot{\psi}_E$
3.  $\mathcal{R}_2$  is the navigation frame, ego-centered and pointing toward the sky
4.  $\mathcal{R}_3$  is the ego-centered frame of the robot.

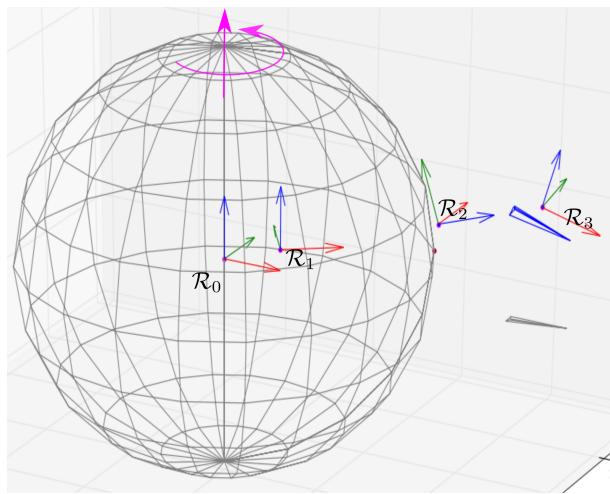


Figure 1.20: Earth with the different frames. The origin of the frames have been shifted for more visibility

For the simulation, take  $\dot{\psi}_E = 0.2 \text{ rad/s}$ ,  $\rho_E = 10\text{m}$ .

2) Provide the kinematic state equations for the body. The state variables are taken as  $(\mathbf{p}, \mathbf{R}_3, \mathbf{v}_3)$ , where  $\mathbf{p}$  is the position of the body in  $\mathcal{R}_0$ ,  $\mathbf{R}_3$  is the orientation of the body, and  $\mathbf{v}_3$  is the speed of the body seen from  $\mathcal{R}_0$  and expressed in  $\mathcal{R}_3$ . The inputs are chosen as  $\boldsymbol{\omega}_3$ , the rotation vector of the body and  $\mathbf{a}_3$ , the measured acceleration, both expressed in  $\mathcal{R}_3$ . Propose a simulation with the Earth rotating and the body moving in the space with a gravity given by  $\mathbf{g}(\mathbf{p}) = -9.81 \cdot \rho_T^2 \frac{\mathbf{p}}{\|\mathbf{p}\|^3}$ .

3) The planet is now assumed to be made of water and the body is an underwater robot equipped with actuators (such as propellers) to move inside the water. The gravity is taken as  $\mathbf{g}(\mathbf{p}) = -\frac{9.81}{\rho_T} \mathbf{p}$ . The water of the rotating planet drags the robot through friction forces (tangential and rotational). Moreover, the robot which has the same density as the surrounding fluid receives an Archimedes upward buoyant force. Propose a dynamical model describing the motion of the robot. The inputs

will be the rotational acceleration  $\mathbf{u}_{\omega_3}$  et the tangential acceleration  $\mathbf{u}_{a_3}$  expressed in  $\mathcal{R}_3$ . These acceleration result from the action of the actuators creating forces and torques on the robot. Propose a simulation illustrating all these effects.

4) The robot behaves like a 3D Dubins-like car model. More precisely, it is equipped with a single propeller which push the robot forward without any possibility to control the speed. We take  $\mathbf{u}_{a_3} = (1, 0, 0)^T$ . We assume that the robot can chose its rotation vector using rudders, jet pumps or inertial wheels. Find a controller such that the robot goes to East along the equator. Validate by a simulation.

#### EXERCISE 1.20.– Modeling and control of a torpedo

See the correction video at <https://youtu.be/4kLr7KnyNi8>

We consider the autonomous underwater vehicle (AUV), a *Riptide*, represented by Figure 1.4.2. For its locomotion, the robot uses three fins and one propeller. The inputs are  $u_0$ , the rotation speed of the propeller and the angles  $u_1, u_2, u_3$  of the fins. We assume that we have no *side slip* effect. This means that the velocity vector  $\mathbf{v}_r$  has always the direction of the robot, or equivalently, the robot has no lateral speed.

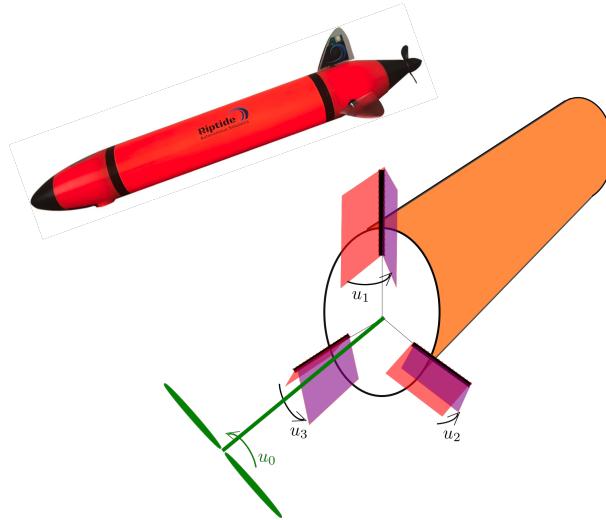


Figure 1.21: The Riptide robot has one propeller and three fins

1) Recall that the mechanization kinematic equations of a solid body (see Equation 1.13) are given by

$$\left\{ \begin{array}{l} \dot{\mathbf{p}} = \mathbf{R}(\varphi, \theta, \psi) \cdot \mathbf{v}_r \\ \dot{\mathbf{v}}_r = \mathbf{a}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \left( \begin{array}{c} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{array} \right) = \left( \begin{array}{ccc} 1 & \tan \theta \sin \varphi & \tan \theta \cos \varphi \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{array} \right) \cdot \boldsymbol{\omega}_r \end{array} \right.$$

where  $\varphi, \theta, \psi$  are the Euler angles,  $\omega_r$  is the rotation vector expressed in the robot frame,  $\mathbf{a}_r$  is the acceleration vector in the robot frame, and  $\mathbf{p}$  is the position of the robot in the world frame. Find a kinematic state space model for the robot, where the inputs are  $\omega_r$  and the longitudinal acceleration  $a_{rx}$ .

2) Considering that the robot has no side slip, propose a dynamic model for the robot. We will assume that the robot has a cylinder shape, that its buoyancy is neutral, and the added mass is equal to the mass of the robot. The inputs are now,  $u_0, u_1, u_2, u_3$ .

3) Propose a controller for the dynamical model to have the equation of the kinematic model. More precisely, we want that the closed loop system has an input to have the desired rotation vector  $\bar{\omega}_r$  and the desired acceleration  $\bar{a}_{rx}$ .

4) Illustrate the behavior of the system using a simulation. Try to get a trajectory similar to that illustrated by Figure 1.4.2.

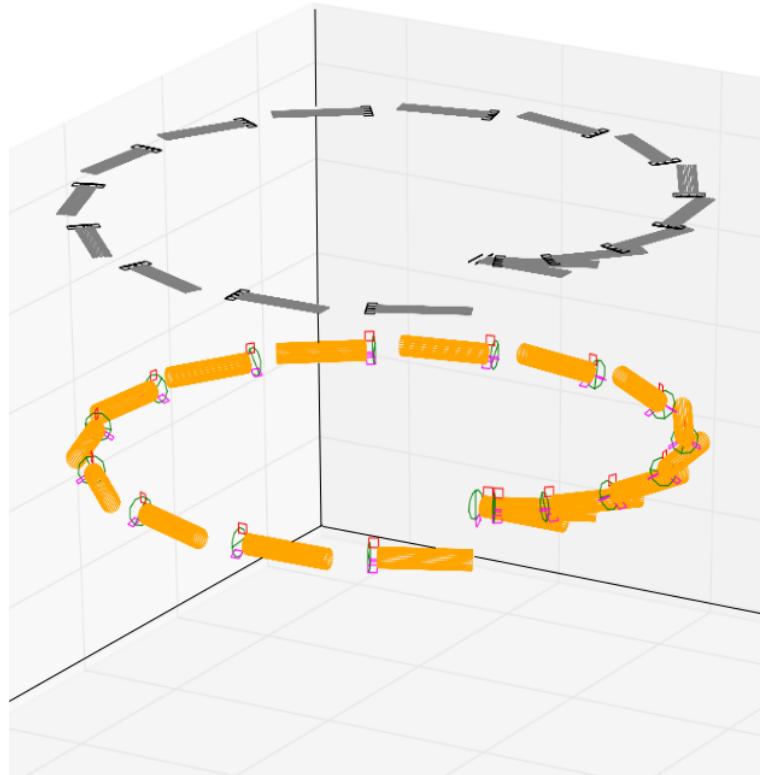


Figure 1.22: The Riptide (orange) is controlled to turn left with a desired speed of  $1ms^{-1}$ . The surface shadow at the surface is painted gray.

---

#### EXERCISE 1.21.– Geodesic

See the correction video at <https://youtu.be/S1-j554Szac>

Consider a  $n$ -dimensional manifold embedded into a  $m$ -dimensional Euclidean space with the parametric equations

$$\mathbf{x} = \mathbf{r}(\mathbf{q}), \mathbf{q} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^m$$

where  $\mathbf{q}$  represents the configuration vector (or the vector of the degrees of freedom) of a robot and  $\mathbf{x}$  is the workspace. We assume that  $n < m$ .

1) Consider a curve  $\mathbf{x}(t) = \mathbf{r}(\mathbf{q}(t))$ . Give an expression of  $\ddot{\mathbf{x}}$  as a function of  $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ .

2) We want  $\ddot{\mathbf{x}} = \mathbf{u}$  where  $\mathbf{u}$  is the wanted acceleration. Now, due to the fact that  $\mathbf{x}(t)$  should stay on the manifold  $\mathbf{r}(\mathbb{R}^n)$ , all accelerations  $\ddot{\mathbf{x}}$  are not allowed. Only the projection of  $\ddot{\mathbf{x}}$  on the tangent plane can be chosen as desired, as illustrated by Figure ?? where the car can turn but should stay on the surface. This means that we cannot have any impact on the component of  $\ddot{\mathbf{x}}$  which is orthogonal to the manifold.

Which  $\ddot{\mathbf{q}}$  should we chose in order to have  $\|\ddot{\mathbf{x}} - \mathbf{u}\|$  as small as possible?

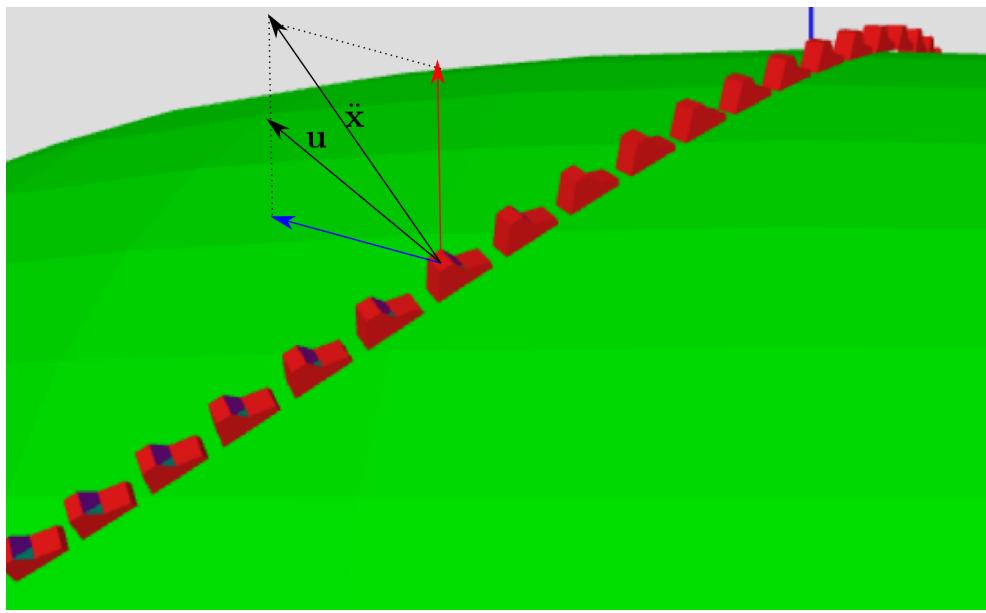


Figure 1.23: When moving on the manifold, we can only have an action the acceleration along the manifold (blue) but not on the transverse component (red)

3) When we do not want any acceleration along the manifold, we chose  $\mathbf{u} = \mathbf{0}$  and the corresponding trajectory is called a *geodesic*. What is the equation of a geodesic starting  $\mathbf{q}(0) = \mathbf{q}_0$  and  $\dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0$ .

4) We consider the torus, given by

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{r}(q_1, q_2) = \begin{pmatrix} (a_1 + a_2 \cos q_1) \cos q_2 \\ (a_1 + a_2 \cos q_1) \sin q_2 \\ a_2 \sin q_1 \end{pmatrix}.$$

with  $a_1 = 10, a_2 = 6$ . Compute and draw a geodesic on the torus starting from  $\mathbf{q}(0) = (0, 0)$  and  $\dot{\mathbf{q}}(0) = (0.1, 0.2)$ ,  $t \in [0, 50]$ .

5) We consider the ellipsoid, described by

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{r}(q_1, q_2) = \begin{pmatrix} a_1 \cos q_2 \cos q_1 \\ a_2 \cos q_2 \sin q_1 \\ a_3 \sin q_2 \end{pmatrix}$$

where  $a_1 = 8, a_2 = 4, a_3 = 2$ . Compute a geodesic starting from  $\mathbf{q}(0) = (0, 0)$  and such that  $\mathbf{q}(10) = (2, 1)$ .



# Bibliography

- [BEA 12] BEARD R., McLAIN T., *Small Unmanned Aircraft, Theory and Practice*, Princeton University Press, 2012.
- [BOY 06] BOYER F., ALAMIR M., CHABLAT D., KHALIL W., LEROYER A., LEMOINE P., Robot anguille sous-marin en 3d, *Techniques de l'Ingénieur*, 2006.
- [CHE 07] CHEVALLEREAU C., BESSONNET G., ABBA G., AOUSTIN Y., *Les robots marcheurs bipèdes ; Modélisation, conception, synthèse de la marche, commande*, Hermès-Lavoisier, Paris, 2007.
- [COR 11] CORKE P., *Robotics, Vision and Control*, Springer, Berlin Heidelberg, 2011.
- [CRE 14] CREUZE V., Robots marins et sous-marins ; perception, modélisation, commande, *Techniques de l'ingénieur*, 2014.
- [DEL 93] DE LARMINAT P., *Automatique, commande des systèmes linéaires*, Hermès, Paris, France, 1993.
- [DRE 11] DREVELLE V., *Etude de méthodes ensemblistes robustes pour une localisation multisensorielle intègre, Application à la navigation des véhicules en milieu urbain*, PhD dissertation, Université de Technologie de Compiègne, Compiègne, France, 2011.
- [DUB 57] DUBINS L.E., On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents, *American Journal of Mathematics*, vol. 79, **3**, p. 497-516, 1957.
- [FAN 01] FANTONI I., LOZANO R., *Non-linear control for underactuated mechanical systems*, Springer-Verlag, 2001.
- [FAR 08] FARRELL, J. *Aided navigation: GPS with high rate sensors*. McGraw-Hill, 2008.
- [FLI 13] FLIESS M., JOIN C., Model-free control, *International Journal of Control*, vol. 86, **12**, p. 2228-2252, 2013.
- [FOS 2002] FOSSEN T., *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics, 2002.

- [KAI 80] KAILATH T., *Linear Systems*, Prentice Hall, Englewood Cliffs, 1980.
- [GOR 11] GORGUES T., Ménage O., TERRE T., GAILLARD F., An innovative approach of the surface layer sampling, *Journal des Sciences Halieutique et Aquatique*, vol. 4, p. 105-109, 2011.
- [MUR 89] MURATA T., *Petri nets : properties, analysis and applications*. Proceedings of the IEEE, vol. 77, 4, 1989.
- [HER 10] HERRERO P., JAULIN L., VEHÍ J., SAINZ M.A., Guaranteed set-point computation with application to the control of a sailboat, *International Journal of Control Automation and Systems*, vol. 8, 1, p. 1-7, 2010.
- [JAU 05] JAULIN L., *Représentation d'état pour la modélisation et la commande des systèmes (Coll. Automatique de base)*, Hermès, London, 2005.
- [JAU 10] JAULIN L., Commande d'un skate-car par biomimétisme, *CIFA 2010*, Nancy, France, 2010.
- [JAU 12] JAULIN L., LE BARS F., An Interval Approach for Stability Analysis; Application to Sailboat Robotics, *IEEE Transaction on Robotics*, vol. 27, 5, 2012.
- [JAU 13] JAULIN L., *Automation for Robotics*, ISTE editions, 2013.
- [JAU 02] JAULIN L., KIEFFER M., WALTER E., MEIZEL D., Guaranteed Robust Nonlinear Estimation with Application to Robot Localization, *IEEE Transactions on systems, man and cybernetics; Part C Applications and Reviews*, vol. 32, 2002.
- [KAL 60] KALMAN E.R. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mex.*, vol. 5, p. 102-119, 1960.
- [KHA 02] KHALIL H.K., *Nonlinear Systems*, Third Edition, Prentice Hall, 2002
- [KLE 06] KLEIN E.M.V., *Aircraft System Identification: Theory And Practice*, American Institute of Aeronautics and Astronautics, 2006.
- [LAR 03] LAROCHE B., MARTIN P., PETIT N., *Commande par platitude, Equations différentielles ordinaires et aux dérivées partielles*, Available at <http://cas.ensmp.fr/~petit/ensta/main.pdf>, 2003.
- [LAT 91] LATOMBE J., *Robot Motion Planning*, Kluwer Academic Publishers, Boston, MA, 1991.
- [LAU 01] LAUMOND J., *La robotique mobile*, Hermès, Paris, France, 2001.
- [LaV 06] LAVALLE S., *Planning algorithm*, Cambridge University Press, 2006.
- [ROM 12] ROMERO-RAMIREZ M., *Contribution à la commande de voiliers robotisés*, PhD dissertation, Université Pierre et Marie Curie, France, 2012.

- [SLO 91] SLOTINE J.J., LI W., *Applied nonlinear control*, Prentice Hall, 1991.
- [SPO 01] SPONG M., CORKE P., LOZANO R. *Nonlinear control of the Reaction Wheel Pendulum*, *Automatica*, vol. 37, p. 1845-1851, 2001.
- [THR 05] THRUN S., BUGARD W. and FOX D., *Probabilistic Robotics*, MIT Press, Cambridge, 2005.
- [WAL 14] WALTER E., *Numerical Methods and Optimization ; a Consumer Guide*, Springer, London, 2014.

# Index

- actuator, 4
- atan2, 15
- AUV, 33
- bank, 14
- Cardan angles, 13
- cross product, 9
- dead reckoning, 13
- Denavit-Hartenberg parametrization, 30
- direction cosine matrix, 12
- elevation, 14
- Euler angles, 13
- Euler's second law, 20
- GESMA, 31
- heading, 14
- homogeneous coordinates, 32
- inertial unit, 17
- intelligence, 4
- Jacobi identity, 23
- kinematic model, 19
- manipulator robot, 30
- mobile robot, 4
- mobile robotics, 4
- modeling, 5
- pitch, 14
- quadrotor, 20
- Redermor, 31
- Rodrigues' formula, 24
- roll, 14
- rotation matrix, 6
- rotation vector, 8
- Runge Kutta, 27
- Schuler period, 27
- sensor, 4
- SNAME, 15
- special orthogonal group, 7
- state representation, 5
- Staubli, 30
- Varignon's formula, 24
- wheel, 35
- yaw, 14