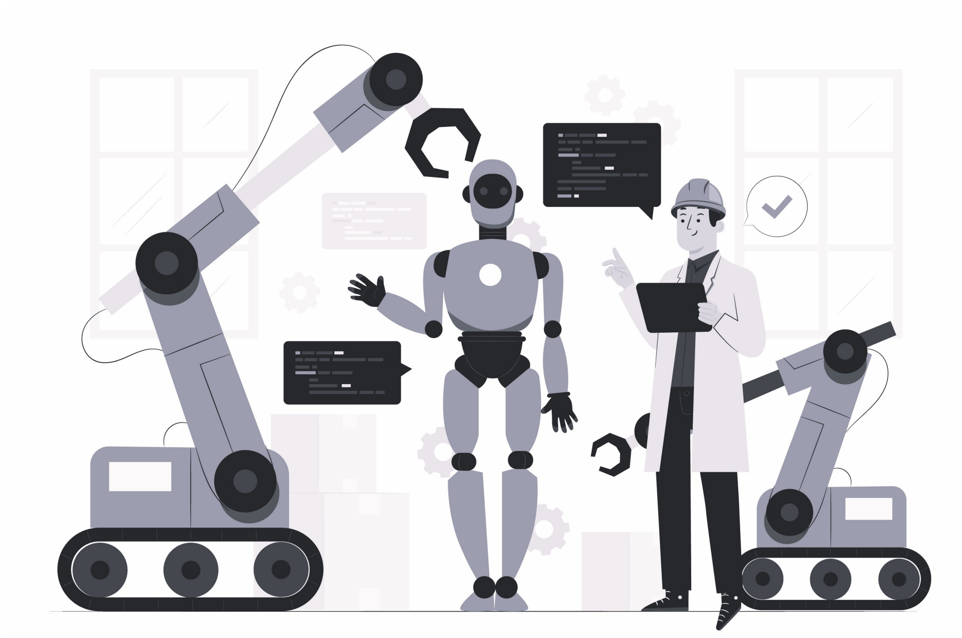
Politecnico di Torino

ROBOTICS

Lecture notes



Carlo Migliaccio
Master's Degree in Computer Engineering (LM-32)

Academic Year 2024/25

Contents

Chapter 1

Introduction

In this chapter we will see...

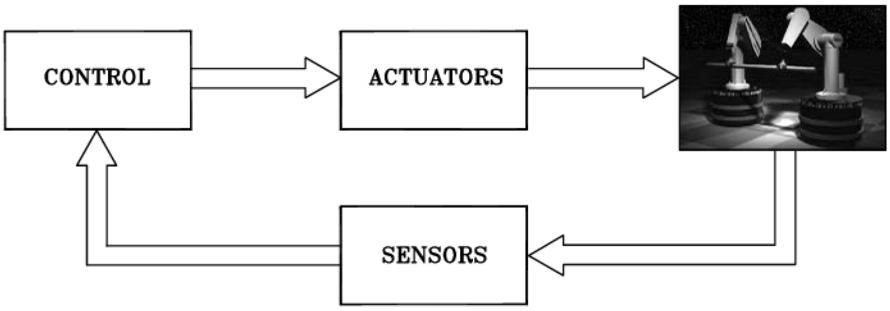


Figure 1.1: Components of a robotic system

1.1 Robot: a possible definition

What is robotics? Before answering this question, it makes sense to rise up another question: **what is a robot?** Among all the available definitions the most suitable is the one provided by Mike Brady:

A robot realizes the *intelligent* connection between **perception** and **action**.

With reference to such a definition we can dissect it finding useful insights. A *robotic system* is in the reality a complex system which is functionally represented by **multiple subsystems**. In particular you have a *mechanical systems* which is made up of two apparatus: **locomotion apparatus** and a **manipulation apparatus** by which the robot itself can carry out the task. Both these subsystems are provided with an *actuation system* which **animates** the mechanical part (this includes *servomotors,drives* and *transmissions*)[1] The **perception** part rely on sensors which can acquire data about either the <u>internal status</u> of the mechanical system (*proprioceptive sensors*, eg. position transducers) or the <u>external status</u> of the environments (they are called *esteroceptive sensors*, eg. cameras).

[1]It is quite clear that the **action** part is linked to the actuation system

The connection between the the actuation and sensing part is provided by a *control system* which in an intelligent way can command the execution of some actions. It is remarkable that the control system exploits a mathematical model of the robot.
At this point, we can say that **Robotics** deals with

study and **design** of robots.

this is the reason why it results in an interdisciplinary subject involving *mechanics, control, computers* and *electronics*.
The term **robot** it is derived from the Slav term *robota* (executive labor), and it appeared several years before any robot could be built!

1.2 Robots classification

Nowadays, robots can be used in different contexts according to which they are split in:

- Industrial robots they are used in the industrial field in order to perform several tasks such as soldering, moving part, cutting parts and so on;
- Humanoid and biometric robots they are employed in order to interact with people or animal environments. Not rarely their mechanics is inspired by the real (natural) behaviour.
- Service robots are used in contexts where the actions to be carried out are dangerous for human beings. For example: think about manipulating radioactive matter.
- Exploration robots are used, for example, in the field of space missions to study the atmosphere and the surface of Mars.

Another available classification of robots is the one based on the type of mechanical structure. Is the robot base moving or is it fixed? In the first case we are talking about **mobile robots** (humanoid/biometric, service and exploration robots often fall in this category), in the latter case we refer them with the term of **robot manipulators** (the great majority of industrial robots).

1.2.1 Robot manipulator (Industrial robots)

We anticipate here that a *robot manipulator* is made up of a sequence of rigid bodies (links) connected by means of articulations (*joints*); the part of the manipulator that ensures mobility is the so-called **arm**, the part devoted to its agility (more technically, dexterity) is the **wrist** finally the part which perfors the action is the **end-effector**. Several types of robot manipulators are obtained changing such components.

1.2.2 Mobile robots

The main feature here is the presence of a *mobile base* which allows the robot to move into the environment it is placed. They are provided with a locomotion system that can be different (wheels, legs, wings...).

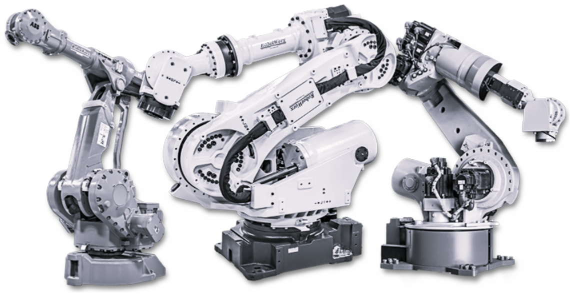


Figure 1.2: Examples of Industrial Robots

1.3 Robot modeling, planning and control

The totality of the tasks to be completed requires the execution of a *specific motion prescribed to the robot*. The correct execution of the motion is entrusted to the control systems which should provide actuators with **command** consitstent with the desired motion (motion control), this – in turn – rely on an accurate analysis on the features of (jointly) mechanical structrure, actuators and sensors. The final aim of such an analysis is the derivation of a mathematical model describing the I/O relationship for the robot components (modeling component). Relevant topics in *modeling, planning* and *control* are briefly resumed in the following.

1.3.1 Modelling

Obtaining a model for a robot (industrial or mobile) concerns essentially in studying both the **kinematics** and **dynamics**. The former is the description of the robot motion without taking into account forces and torques which causes it. The latter describes forces and torques which generates the motion.
A further distinction is made between between *kinematics* and *differential kinematics* is made. In the first case we want to find a relationship between the joints and end-effector positions. In the second case we want to study the relationship between the joint and end-effector motion by means of velocity. A very useful tool in this case is the *Jacobian* of the manipulator.
We conclude this discussion by saying that having a model for the robot is useful for the mechanical design of the structure, choice of actuators, determination of control strategies and for doing computer-based simulations.

1.3.2 Planning

Every robot performs a task, for example for a manipulator we have the necessity to describe in some way the motion at the joints or at the end-effector. In material-handling task for example it is sufficient to decide an initial position and and end position (*point-to-point* motion) in other situations we have to track a certain trajecotry (*path motion*). In all these cases the task is the **trajectory planning** that is the determination of timing laws for relevant variables. For a mobile robot, such a task is even more complicate since it requires to take into account the constraints imposed by the wheels, legs, wings...

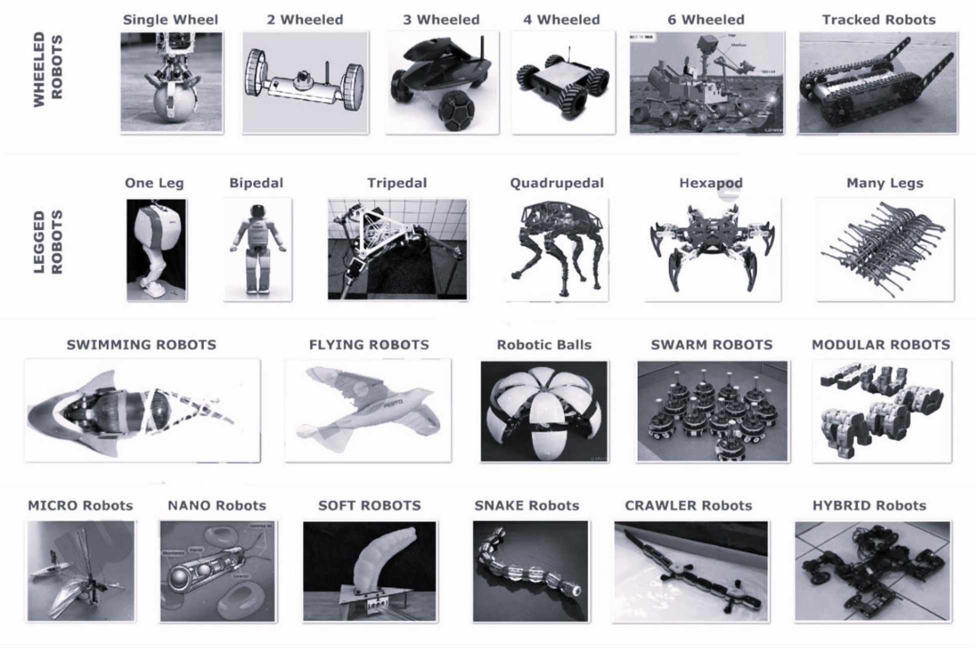
1.3.3 Control

The generated trajectories constitutes the reference signal for the *control system* of the mechanical structure. Techniques for controlling manipulators and mobile robots are very different how we will see this is due the presence or not of the interaction robot-environment and of the mobile base.

Chapter 2

Mobile robots

In this chapter we will see...



2.1 Introduction

Definition 2.1.1 (**Mobile robot**)**.** A **mobile robot** is a structure capable of moving and act in *terrestrial*, *underwater* or *aerial* environments.

The characteristics of the environment are fundamental since planning and control are affected by it. Namely the environment can be totally structured, unstructured or partially structured. We refer the **structuredness** of the environment as the knowledge on geometric characteristics.

2.1.1 Autonomy

A mobile robot is equipped with a certain level of **autonomy** which makes the structure capable of moving independently from a human supervisor. For achieving it, it is required a computational part (CPU, Intelligence of the robot), some sensors and actuators and an energy source, which can be either generated on-board or provided by mean of an external source.

2.1.2 Locomotion

Another fundamental aspect is that, differently from manipulators, a mobile robot is equipped with a **locomotion apparatus** which drastically changes according to the environment they are acting in. A **terrestrial robot** can have wheels, legs or a biomimetic locomotion system; a **underwater robot** can provided with propellers or water jets; finally, an **aerial robots** can have rotating, fixed or flapping wings.

2.2 Types of wheels

In the following we are focusing our attention on the wheeled robots, as they represent the great majority of mobile robots used in applications. The basic mechanical structure of this robot is indeed the wheel. The table Table 2.1 shows a summary of the fundamental information about the most popular types of wheels.

2.3 Constraints and Kinematic Models

We have seen in the introduction that mobile and industrial robots have different characteristics, this results in different models, planning and control strategies. Since in general robots are made up of multiple rigid bodies (*multibody system*) linked together by joints, their motion is constrained. This holds in general, however also the types of constraints are different in the case of mobile robots and manipulators. This is because in the former case you have limitations in the **position** of the whole body, in the case of mobile robots, and in particular in terrestrial wheeled robots, you have instead limitations on **how the position can change** in magnitude, direction and side (this is nothing but the **velocity**). From now on we are focusing our attention on **modeling wheeled robots**.

2.3.1 Constraints and their classification

In order to properly describing and giving a classification of constraints, it is better recalling the concept of **generalized coordinates**.
Let the vector $\mathbf{q} \in \mathbb{R}^n$ be the *generalized coordinates* that describes the **configuration** of the robot (minimum number of variables needed to model the robot motion). For the moment, let us assume that the **configuration space** $\mathcal{C}$ coincides with $\mathbb{R}^n$. For example for a **unicycle** the generalized coordinates are in number of three:

$$\mathbf{q} = [x \quad y \quad \theta]^T \in \mathbb{R}^3 \tag{2.1}$$

where x and y are the position of the contact point between the (single) wheel and the plane on which the motion occurs, while θ is the angle of the wheel with respect to the horizontal axis. The **evolution in time** of the generalized coordinates vector *describe the motion* of the system.

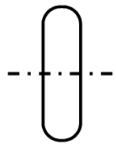
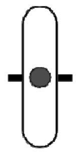
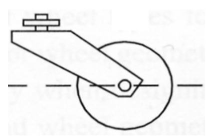
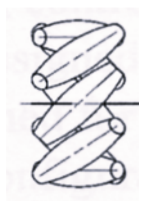
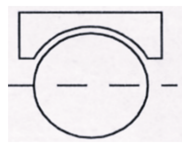
WHEEL TYPE	**DESCRIPTION**	**SYMBOL**
SIMPLE NON-STEERING WHEEL	They can rotate about an axis which passes through the center of the wheel itself, orthogonal to its plane. The orientation of the chassis with respect to the wheels is constant.	
SIMPLE STEERING WHEELS	Has two axes of rotation, one that is orthogonal to the wheel plane, the other which is *vertical* and goes through the center of the wheel. This provides the wheel with the possibility of changing the orientation with respect to the chassis. Note that for both non-steering and simple steering wheels the component the velocity which orthogonal to the wheel plane is null since there is no sleeping. That is: $$v^{\perp}(t) = 0$$	
CASTOR WHEEL	It is a variant of the previous one in which the vertical axis does not pass through the center of the wheel from which it is **displaced** by a constant offset. This adds degrees of freedom to the vehicle on which they are mounted on. Such a type of wheels are often used for office chairs and supermarket carts.	
OMNIDIRECTIONAL OR SWEDISH WHEEL	There is another type of non-conventional wheel that is the *mechanum* (or **swedish wheel**). It mounts some passive rollers whose rotation axis is inclined by 45 degrees with respect to the plane of the wheel itself. They are also called *omniwheels*.	
SPHERICAL OMNIWHEEL	There is another type of omniwheel which is **spherical**. They can be either active or passive. Like in the case of swedish wheels, a vehicle equipped with four of them is called **omnidirectional**.	

Table 2.1: Types of wheels with description and symbols

Constraints can be found by mean of **equality** (in this case we refer them as *bilateral* constraints), by mean of **inequality** (we refer them as *unilateral* constraints). Furthermore, according to the fact they are or not time-variant, we can divide them in *rehonomic* (explicit dependence on time) and *scleronomic* (time-invariant constraints). In this course we will treat only **bilateral and scleronomic constraints** another term for indicating them is **holonomic** (or **integrable**) constraints.

Holonomic constraints

Such a type of constraint[1] can be expressed as:

$$h_i(\mathbf{q}) = 0, \quad i = 1, ..., k < n \tag{2.2}$$

A system whose motion is characterized only by holonomic constraints is called **holonomic system**. By using the *implicit function theorem* (or in Italian "Teorema del Dini"), we can reduce the dimension of the configuration space to $n - k$

Kinematic constraints

Such a type of constraints involve both generalized coordinates $\mathbf{q}$ and its derivative $\dot{\mathbf{q}}$. In the most general case they can be expressed as:

$$a_i(\mathbf{q}, \dot{\mathbf{q}}) = 0, \quad i = 1, ..., k < n \tag{2.3}$$

Kinematic constraints are limiting the set of generalized velocities that can be obtained by each configuration. In some cases they can be written in the so-called **Pfaffian form**, that is they can be written as a *linear combination of the generalized velocities* $\dot{\mathbf{q}}$:

$$\mathbf{a}_i^T(\mathbf{q})\dot{\mathbf{q}} = 0, \quad i = 1, ..., k < n \tag{2.4}$$

An example of kinematic constraint in Pfaffian form is:

$$3q_1\dot{q}_1 + 2\sin q_1\dot{q}_2 + \sin q_3\dot{q}_3 = 0$$

The reason for using such a notation is that we can immediately retrieve the expression of the associated function by doing the row-by-column product (standard inner product). Such k kinematic constraints can be written in compact form by introducing matrices

$$\mathbf{A}^T(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \tag{2.5}$$

where $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{k,n}$. It is interesting to note that the presece of k holonomic constraints imply the presence of k kinematic constraints. This can be easily showed by computing the time derivative for the k holonomic constraints:

$$\frac{dh_i(\mathbf{q})}{dt} = \frac{dh_i(\mathbf{q})}{d\mathbf{q}} \cdot \frac{d\mathbf{q}}{dt} = \frac{dh_i(\mathbf{q})}{d\mathbf{q}} \cdot \dot{\mathbf{q}} = 0, \qquad i = 1, ..., k \tag{2.6}$$

where we have applied the fact that the constraints are holonomic while using the *Chain rule* for passing from the first to the second step. From the Equation (2.6) we have understood that:

$$\text{holonomic constraint} \Longrightarrow \text{kinematic constraints}$$

[1]We recall that for a multibody system described by using the Lagrangian approach, holonomic constraints are those allowing a reduction of the number of needed variables.

In general, we cannot say the inverse and in this case (since the step from the derivative to the primitive results in doing the integral) associated constraints are called **nonholonomic** (or **non-integrable**) ones. A system characterized by such constraints is called **nonholonomic system**. In presence of non-integrable constraints the dimension of the configuration space $\mathcal{C}$ cannot be reduced while the generalized velocities can be described over a subspace of dimension $n-k$ (how we are going to see in a minute).

Example of nonholonomic constraint

A unicycle rolls on a plane **without slippering**, we have already seen its generalized coordinates. For such a system we have the so called **pure rolling constraint**, this imply the velocity of the contact point not to have a non-zero component along the direction orthogonal to the wheels plane. By using simple trigonometric properties involving the inifinitesimal increment dx and dy, we can state that

$$\frac{dy}{dx} = \tan\theta \tag{2.7}$$

Can we obtain the Pfaffian form for such a constraint? The answer is YES. In fact by dividing and multiplying for an infinitesimal time increment dt we obtain:

$$\frac{dy}{dx} = \frac{dy}{dt}\cdot\frac{dt}{dx} = \frac{\dot{y}}{\dot{x}} = \frac{\sin\theta}{\cos\theta} \iff \dot{y}\cos\theta = \dot{x}\sin\theta$$

Which is the same to say that

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \iff [\sin\theta \quad \cos\theta \quad 0]\dot{\mathbf{q}} = 0 \tag{2.8}$$

The constraint we have just derived can be demonstrated that is non-integrable and so nonholonomic, in fact we cannot reduce the dimension of the configuration space (in other words all of the generalized coordinates are needed to properly describe the unicycle motion). Note that, start from an initial state $\mathbf{q}_i$, you can bring the system to any final state $\mathbf{q}_f$, under the assumption of not to violating the pure rolling constraint[2].

2.3.2 Kinematic model

From the Equation (2.5), we can immediately see that the $n-k$ admissible generalized velocities belong to the null space[3] of $\mathbf{A}(\mathbf{q})^T$ that is

$$\dot{\mathbf{q}} \in \mathcal{N}(\mathbf{A^T}(\mathbf{q})) \tag{2.11}$$

[2]We will see that in order to pass from an initial to a final state a *trajectory planning algorithm* must be used.

[3]Just for doing a brief recap. Given a matrix $A \in \mathbb{R}^{m,n}$ we can individuate the following sets (vector spaces):

- Null space that is the a subset of $\mathbb{R}^n$ defined as:

$$\mathcal{N}(A) = \{x \in \mathbb{R}^n :\ Ax = 0\} \tag{2.9}$$

- Range space that is a subset of $\mathbb{R}^m$ defined as:

$$\mathcal{R}(A) = \{y \in \mathbb{R}^m :\ y = Ax\} \tag{2.10}$$

  the dimension of such a vector space is called the rank of the matrix A (rank(A)) and it holds that its maximum value is the $\min(m, n)$.

Another important result is that

$$n = \dim\mathcal{N}(A) + \dim\mathcal{R}(A)$$

Knowing n (for us dimension of the configuration space) and the rank of A, we can find the dimension for the null space of A.

We know that this is a vector space and it has got a basis of $n-k$ elements which we can denote with $\{\mathbf{g}_i(\mathbf{q})\}_{i=1}^{n-k}$, we can group together such elements in a matrix $\mathbf{G}(\mathbf{q})$ so that the generalized velocities can be expressed as

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{u} \tag{2.12}$$

this is nothing but the **kinematic model** of the constrained system (a system of ordinary differential equations[4]). The vector $\mathbf{q}$ is called the *state vector* while $\mathbf{u}$ is the **input vector**. Moreover the obtained system is said to be driftless since in absence of an input the generalized velocity is null. Not rarely the components u_i of $\mathbf{u}$ have a meaning related to the physics or the available control input.

In the following for better fixing the concepts we have just given, two example of kinematic models are given.

Kinematic model for the unicycle

Let us consider, a bit more in details, the unicycle system. It is noticeable that the line in which the motion does not occur is called **zero motion line**.
We have said that the generalized coordinated $\mathbf{q}$ are the ones in Equation (2.1). We have a single constraint that we have reduced in Pfaffian form. The next step *in order to obtain a kinematic model* is determining a base for the null space of the constraint matrix. One possible choice for vector fields $\mathbf{g_i}(\mathbf{q})$ is[5]:

$$\mathbf{g}_1(\mathbf{q}) = \begin{bmatrix}\cos\theta & \sin\theta & 0\end{bmatrix}^T, \quad \mathbf{g}_2(\mathbf{q}) = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$$

Therefore, putting them together we obtain the matrix

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix}\cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1\end{bmatrix} \tag{2.13}$$

The kinematic model for the unicyle, then, can be written as:

$$\begin{bmatrix}\dot{x} \\ \dot{y} \\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta \\ \sin\theta \\ 0\end{bmatrix} v + \begin{bmatrix}0 \\ 0 \\ 1\end{bmatrix} \omega \tag{2.14}$$

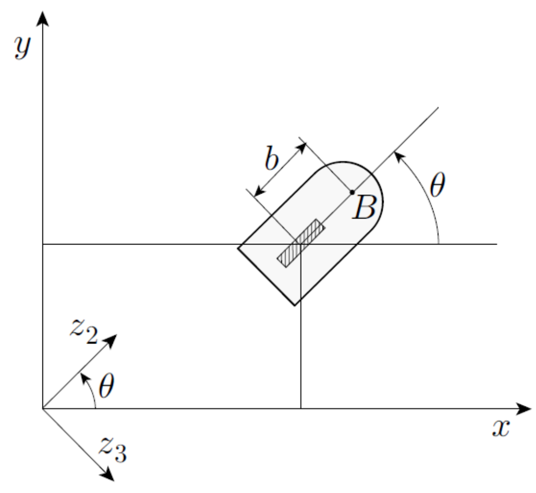


Figure 2.1: Choice of generalized coordinates for the unicycle

In such a context the elements of the input vector have a physical meaning, since v is the driving velocity, while ω is nothing but the steering velocity. Why are we interested in studying the kinematic model of a unicyle? Consider that without a human beings balancing the system this is also an unstable system! However, there are some robot (stable) robot structures that from a **kinematic point of view** are equivalent to unicycles. We are talking about *differential drive* and *synchro drive* vehicles.

[4]ODE

[5]Note that this is a vector field since both domain and codomain are vectors of suitable dimensions.

Differential drives: a stable structure for the unicycle model

A *differentially driven robot* has two independent wheels with different angular velocities ω_R and ω_L. Both wheels have a radius r and are constrained to be at a distance d. A third wheel is passive, in the sense that there is not a motor changing that velocity. By doing a proper choice for the control input, we can obtain a kinematic model that is totally equivalent to the one of a unicycle. In particular, from the left and right wheels angular velocities we can obtain:

$$v = \frac{r}{2}(\omega_R + \omega_L) \tag{2.15}$$

$$\omega = \frac{r}{d}(\omega_R - \omega_L) \tag{2.16}$$

Kinematic model for a bicycle

A **bicycle** is vehicle with a steered wheel and a fixed one, the distance between the wheels is fixed to be L. As usual we have to choose the (generalized) coordinates for such a vehicle a possible choice is the following:

- (x, y) being the contact point of the rear[6];
- θ is the angle between the rear wheel and the x axis (this is nothing but the orientation of the vehicle with respect to the x axis);
- ϕ is the steering angle of the front wheel.

Remark. It is interesting focus our attention on a point: why do not we introduce also the coordinates of the front wheel? Well, it is sufficient having a single contact point since, back and front wheel are at a fixed distance ℓ, and this is nothing but an *holonomic constraint* which allows us the shrinking of the configuration space dimension!

Going on into the discussion, there are essentially two *pure rolling constraints*, one for each wheel:

$$\dot{x}_f \sin(\theta + \phi) - \dot{y}_f \cos(\theta + \phi) = 0 \tag{2.17}$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \tag{2.18}$$

while indicating with (x_f, y_f) the coordinates of the front wheel center, while $(\theta + \phi)$ is its angle with respect to the fixed reference frame. We have just said that they are not strictly necessary, since they can be obtained starting from the coordinates of the other wheel in particular:

$$\begin{cases} x_f = x + \ell \cos\theta \\ y_f = y + \ell \sin\theta \end{cases} \tag{2.19}$$

using the trigonometry, the Equation (2.17) can be expressed as

$$\dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - \ell\dot{\theta} \cos(\phi) = 0 \tag{2.20}$$

where we have used that $\sin^2\theta + \cos^2\theta = 1$ and the trigonometric formulas for $\cos(\theta + \phi)$ and $\sin(\theta + \phi)$. The derived constraints can be put if Pfaffian form using the matrix

$$\mathbf{A}^T(\mathbf{q}) = \begin{bmatrix} \sin\theta & -\cos\theta & 0 & 0 \\ \sin(\theta + \phi) & -cos(\theta + \phi) & -\ell\cos\phi & 0 \end{bmatrix} \tag{2.21}$$

[6] back wheel

Our objective is obtaining a basis for the null space of such a matrix in order to derive the kinematic model. Since $rank(\mathbf{A}^{\mathbf{T}}(\mathbf{q})) = 2$, the dimension of its null space is given by the difference between the dimension of the configuration space and such a rank, that is 2. A possible basis for such a null space is given by the column of

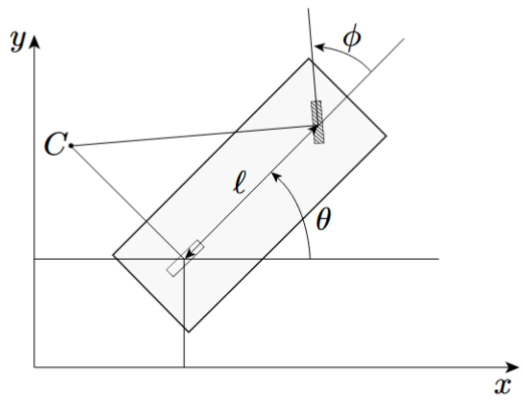


Figure 2.2: Possible choice for the generalized coordinates of a unicycle

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} \cos\theta\cos\phi & 0 \\ \sin\theta\cos\phi & 0 \\ \frac{\sin\theta}{\ell} & 0 \\ 0 & 1 \end{bmatrix} \tag{2.22}$$

The associated kinematic model is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \frac{\sin\theta}{\ell} \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{2.23}$$

the control input u_1 can be choosen according to the drive in particular $u_1 = v$ if the vehicle is front-drive, $u_1 = v/\cos\theta$ if the vehicle is back drive, since the first two equations must be equivalent to the ones of a unicyle. Just for a matter of notation/nomenclature the instersection point C between the two zero motion lines is called *instantaneous center of rotation*, depends only on $\mathbf{q}$ and it is interesting since each point of the chassis is moving instantaneously along along the circumference centered at C (see Figure 2.2).
Equivalent (stable) systems having the same kinematic model are the **tricyle** and the **automobile**.
An interesting point is that, if we assume that the steering velocity can be *manually controlled*, the number of state variables (generalized coordinates) for the bicycle is the same than the unicycle. In this way we can model both systems using the same kinematic model.

Remark. We remind that the kinematic model represent a system of ODE which can be solved using any integration method available in MATLAB. For example we can obtain the trajectory of the unicycle/bicyle model and the time law for θ by using the command `ode45`.

2.4 Sensors for mobile robots

Sensors play a crucial role in the field of robotics and mobile robots. They are, approximately, devices which takes measurement from the environment and convert them in a form which can be further analyzed by the computer of which the robot is equipped.
Before focusing on sensors, let us give the definition of what a **transducer** is. This is a device by a ehich signal from one form of energy (eg. mechanical) is converted into another form of energy. Moreover can be either *emitters* or *receivers.*
Sensors are receivers which covert a measurable signal intol an electric one. Very often in order to be digitally elaborated, such signals are converted by using an ADC device.

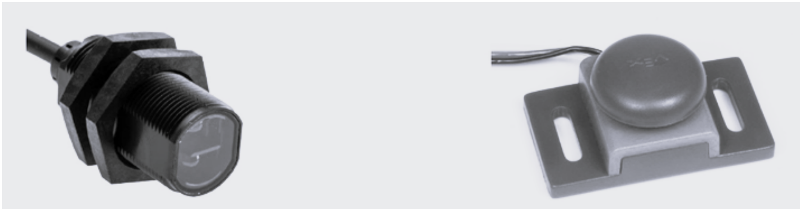


Figure 2.3: Proximity and contact sensors. From left: photoelectric and bumper

In the following, after giving some possible classifications and features about sensors, we introduce the main types of such devices used in the field of robotics and in particular in the field of *mobile robotics*.

2.4.1 Sensors classification and features

There are different classifications can be done about sensors, each one focusing on a different aspect:

- Passive/Active sensors the former ones measure the energy from the environment directly the latter ones at first inject some form of energy (eg. light) and then they measure the effects such energy produces.
- Proprioceptive/Exteroceptive sensors is a classification based on what type of measurements the sensors themselves are taking. A *proprioceptive* sensor takes measures of quantities related to the robot (internal), the other are taking the measurement with reference to the *external world*.
- Type of taken measurement in the sense that there are sensors measuring contact, position, velocity and so on.

In the Table 2.2 we are going to give main characteristics and features for common sensors.

2.4.2 Proximity and contact sensors

They are used from mobile robots in order to sense object which are either in front of them or nearby, at a distance that is fixed or parametrizable. The most general family is known as *proximity sensors*, when the distance is zero (within a certain tolerance), these are *contact sensors*. Such a type of robot can be used for both safety or navigation tasks. The most common forms in which you can find them are:

1. *Photoelectric proximity* Uses a couple LED + Photoresistor in otder to sense the reflected light. It may receive a signal which is proportional to the distance.
2. *Bumpers* Such devices present as *microswitches* attached to the protective case and they are used to receive shocks and then to sense the contact.

One example for both types is shown in Figure 2.3.

2.4.3 Encoders

This are part of the *proprioceptive sensors* and are devices which converts a **linear or angular position** into a digital code. The building block of any type of encoder is a *disc* subvided in sectors embedded with a mechanism to count sectors. They are substancially based on optical mechanism or exploit the Hall effect.

FEATURE	DESCRIPTION
Linearity and nonlinearity	A sensor if the relationship between its input and the related output is linear. Such sensors result (approximately) in simple models, moreover the *effect of noise* is also linear in the sensor range.
Measurement range	A sensor cannot take whatever measurement. In particular, they are in a certain interval with a smallest and a largest value. Namely $Range = [A, B]$
Dynamic range	It is nothing but the ratio between B and A expressed in dB, that is: $$\text{Dynamic range} = 20 \log_{10} \frac{B}{A}$$
Sensitivity	Is nothing but the slope dy/dx of the sensor response. An accurate sensor should have *high sensitivity.* If it is a linear one such a parameter (the derivative) is constant. The sensitivity can be used in order to detect the saturation: you have saturation when to input variation, there is no output variation. Such a phenomenona occurs outside the *measurement range*
Resolution	Is the minimum variation of the measured physical variables that can produce a detectaable change in the ouptut. It can be associate to either physical limitations or ADC process
Precision	It represents the metric about the *reproducibility* of given measurements. Ideally one would have the same measurement again and again, real sensors provide a *range of values* over time which are distributed according to a certain statistical distribution.
Accuracy	It is a metric quantifying the *correctness* of the output provided by a sensor compared to the real value of the measured signal. The concept of *real value* can turn in a philosophical one. For this reason, accuracy is assesses by taking in relation with other more accurate devices.
Bandwidth	It is known as the *maximum frequency* at which the sensor provides reliable measurements. We want the sensor bandwitdth to be sufficiently, but not too much, large so that the noise (tipycally high frequency signal) is avoided.
Response time	Is the time which elapses between the change in the input and the associated variation of the sensor output.

Table 2.2: Sensors characteristics and features

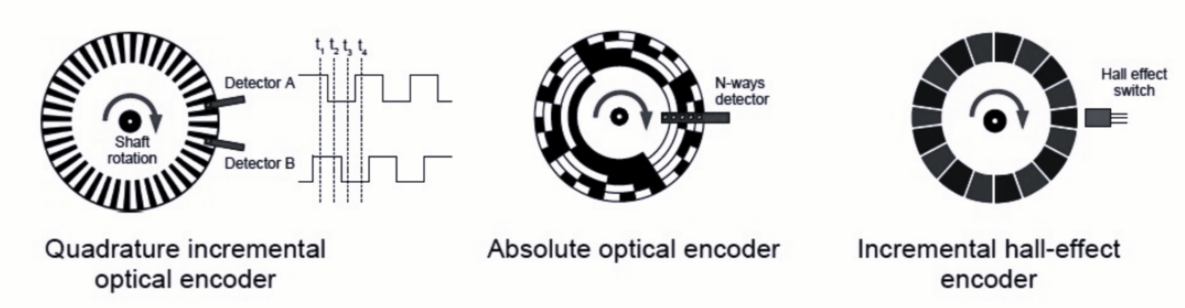


Figure 2.4: Different types of encoder

2.4.4 Global Positioning Systems

In the case of *GPS (Global Positioning System)* the localization is made possible through a set of distinguished landmarks of beacons. Such a type of localization can occur either outdoor (in the nature) or indoor (in a closed place). Different protocols are used in both cases.

Outdoor positioning: GNNS

GNSS stands for *Global Navigation Satellite System Receivers* here a receiver (put on the robot) can sense signals from a subset of **satellites**. Using the time of arrival of such a signal the *distance can be estimated*. Using the trilateration, *latitude, longitude* and *elevation* can be found. This is the basic version of the protocol. However there are some variants:

DPGS (Differential Global Positioning System) The GPS receiver communicates with one or more base stations (on the Earth) in order to correct common error related to the transmission from the satellites to the Earth ground. Accuracy is of the order of meters.

RTK (Real time kinematics) Here instead of analzying only the time of arrival also the carrier wave is used. The distance is inferred from the number of cycles to which is added the phase difference. In particular, the *phase difference* is provided by a base station which serves a certain number of satellites. This technique provides a fine-grained correction of the measurement, in this way the accuracy passes from meters to centimeters.

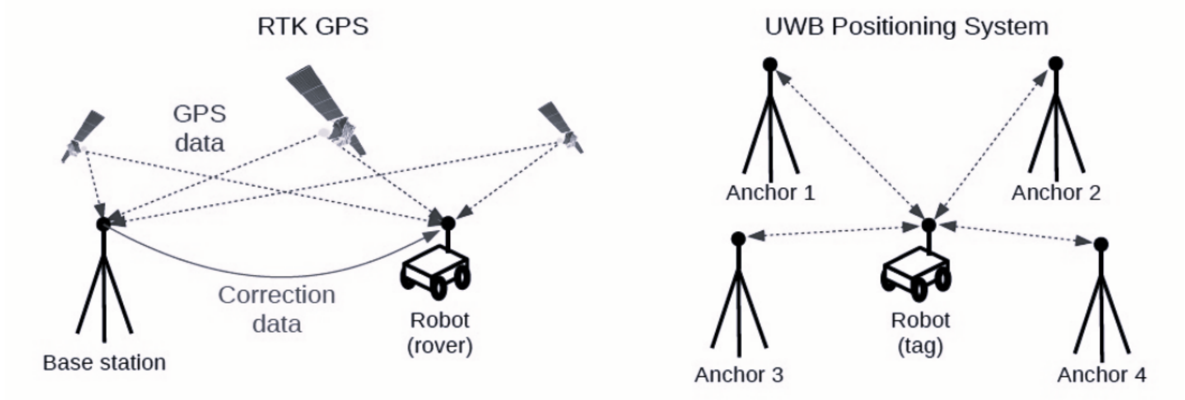
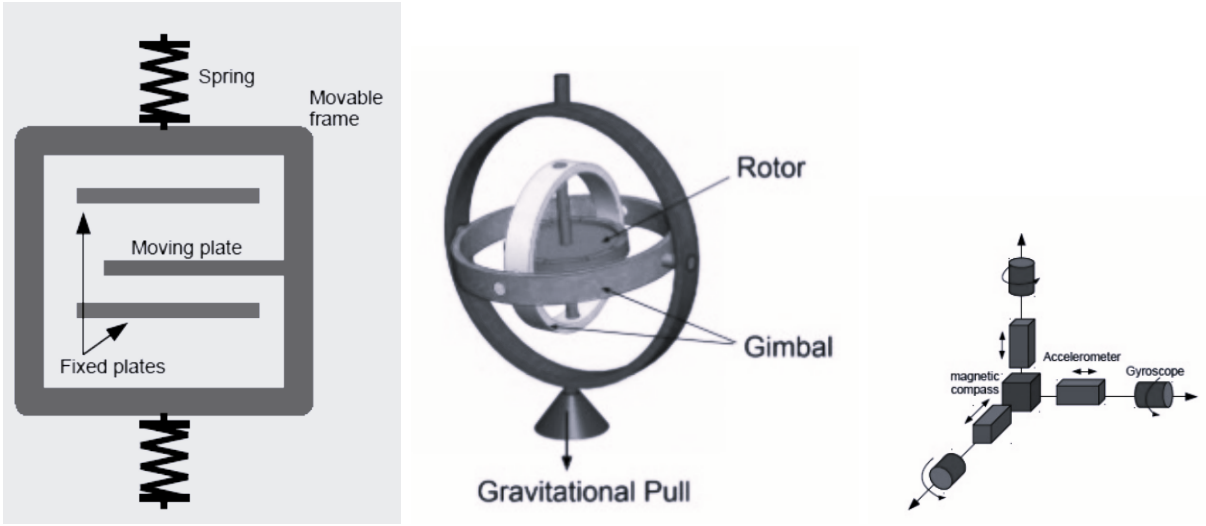


Figure 2.5: Global Positioning System



(a) Accelerometer (b) Gyroscope (c) IMU

Figure 2.6: Inertia and Heading sensors

Indoor positioning: UWB

UWB (Ultra-wideband positioning system) is based on the same concept of GNSS. However there are some differences: (i) some *anchors* are used with a frequency in the range $[3.1, 10.6]$GHz; (ii) The communicaiton is bidirectional and the 3D localization can be achieved by mean of a 3D placement of the anchors. Note that each robot being localized by using UWB must be equipped with a **UWB tag**. Even in this case different strategies are used.

TDOA (Time difference of arrival) In this case the UWB tag constantly broadcast a message with its own identity. The anchors will receive such a signal with *time differences* since the position is different. If the anchors are all synchronized on the same clock signal, is possible performing the localization by using trilateration.

TOF (Time of flight) This technique is applied when there is the absence of a *global clock*. In particular, a 2-way protocol is used: a message is sent by using the UWB tag and a control message is bounced back. The distance from the anchors is obtained by computing the *time of flight*, finally trilateration is applied to retrieve the indoor positioning.

2.4.5 Inertial and Heading Sensors

There are some sensors which use the **inertia**[7] of the body to retrieve position and orientation of the body itself. Such sensors are mainly **accelerometers** and **gyroscopes**.

Accelerometers

Their objective is to measure the *linear acceleration* along a defined axis, such an acceleration is typically converted into either a force or a deformation (mass-spring based). There are both *piezoelectric* and *capacitive* sensors.

[7]It is the property of a body to change its current state of stillness of motion with constant linear or rotational velocities.

Gyroscopes

They are used for measuring the orientation of a body using a vibrating or rotating body which tends to preserve the initial body's orientation. They can assume the following forms:

Rotating Structure Gyroscopes They use a disk which is mounted on a structure which allows the orientation of the body to be unchanged when the whole structure itself is rotated. The orientation is obtained by computing the differences between the starting and final relative position of the body with respect to the rest of the gyroscope. Non-idealities due to friction must be taken into account for more accurate results.

Vibrating Structure Gyroscopes They exploit the *vibration of a body* along the same direction irrespective with the rotation of the sensor itself. The vibration produces a force, whose measurement is able to provide the **rate of rotation**. The absolute orientation is obtained by mean of integration.

Compasses

They are used in order to measure the direction of the Earth's magnetic field (N-S). This can be made using the *Hall effect* or the *Fluxgate* principle. In the former case is exploited the phenomena according which when a conductor is traversed by a current. In presence of a magnetic field a voltage is generated in the orthogonal direction of the current. In the latter case a measurement of a *magnetic flux* is exploited.

Inertia Measurement Units (IMUs)

An IMU is a device which embeds in a single device all of the Inertia sensors we have seen till now. In particular there is a compass and three couples (Accelerometer, Gyroscope) along the three axis of motion. They are often used to improve the performance of the GPS.

2.4.6 Digital cameras

They are particular devices which produces 2D arrays, by mean of an image, containing the measurement of the *visible light* from a 3D scene. You can imagine that, since a 3D scene is transformed into a 2D array, the third dimension is lost. Main components of cameras are: (i) a imaging sensor, (ii) lens to route the light toward the sensor, (iii) an ADC to convert each pixel into a digital value. Ideally, we wish to reproduce the *pinhole model* in which each single 3D point is mapped into a single pixel realizing the so-called *perspective transformation*. There are mainly two types of cameras: (i) *gray level cameras* give as an output a brightness map which is a measure of the intensity of the elctromagnetic radation; (ii) *Color cameras* measure also the wavelength of such a radiation. In particular, they use separate sensors which are sensitive to different wavelenghts.

Omnidirectional cameras

Vanilla cameras provide a limited field of view. *Omnidirectional cameras* solve this issue by achieving a field of view of 180Â° in both horizontal and vertical directions. fisheye lens and mirrors are used. The pinhole model does not hold anymore, the *spherical projection model* is used instead.

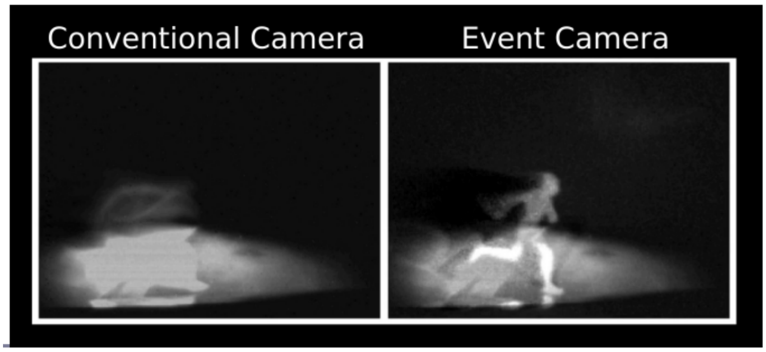


Figure 2.7: Difference between standard and event cameras

Event cameras

Standard cameras provide data at a fixed rate (tipically 50-60 frame per second). *Event cameras* are equipped with asynchronous sensors that generates a series of timestampe events wheter the brightness of the image changes. They are used in task in which is important achieving a certain level of performance.

2.4.7 Ranging sensors

They appears as an extension of proximity sensors, since they provide the *distance to objects*. The output of such cameras is an array of distances (*ranges*) along directions of measurements, or the projection of such distances on the coordinate system of the sensor (*depth*).
A 2D array of range is called a *range map*, on the other hand a 2D array of depth is called a *depth map*. Combing both maps a *point cloud* can be generated.

Sonars

They use *sound waves* to **detect objects** and **measure distances** can be either passive or active. The latter ones are very often used in mobile robots coupled with *ultrasonic waves*. A mobile robot not rarely mounts an array of SONAR sensor in order to cover an angular sector of interest. Also in this case both piezoelectric and capacitive technologies can be exploited.

LiDARs (Light Detection and Ranging)

Such sensors use *infrared lasers* together with reflective properties of the environment in order to measure distances. The generated and received waves can be used in different ways to measure distances.

1. *Continuous wave* (also known as phase shift based) exploit the difference in phase between the generated and backscattered wave.
2. *Pulse Based (PB)* they measure directly the time of flight of a pulse of light, to be interpreted as *round-trip time*.

Improved versions of such vanilla version can correct some non-idealities due to light deflections. They can cover higher ranges wrt to sonars and the accuracy is improved. However, they are very sensitive to environmental factors, finally their cost is much higher.

RADARs (Radio Detection and Ranging)

They use radiowaves (3mm-30cm), are less accurate with respect to LIDAr, but they can use the *Doppler Effect* to measure distances. They cover a range of 200m in detecting position, relative speed and direction of motion. Briefly speaking, they emit a signal called a *chirp*, whose frequency varies linearly over time. The difference between the frequencies of the emitted and received signal varies according to the distance of the object to detect.

ToF cameras

Time of Flight (ToF) cameras represent the meeting point between LIDARs sensors and digital cameras. Here an infrared wave is emitted and its reflected wave is detected by a 2D imaging sensor.

Stereo Cameras

They are based on the concept of disparity, here triangulation is used in order to reconstruct depths. Such cameras can also be passive: here there are two ordinary cameras, the accuracy in this case is lower, moreover untextured homogeneous areas are hard to be measured.

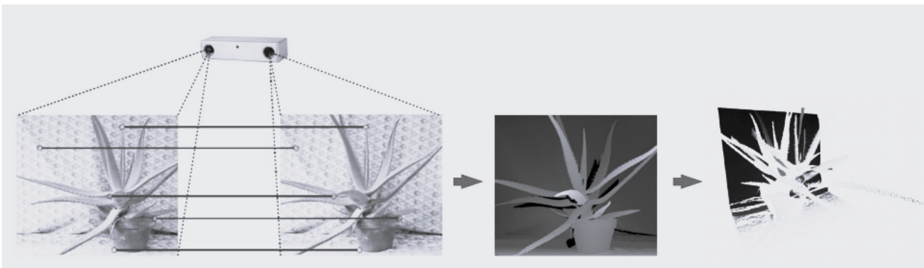


Figure 2.8: Stereo cameras

RGB-D Cameras

They are particular color cameras which also provide the depth estimates. Even in this case they can be both active or passive. Active cameras use a **pattern projector**, which creates a saliency map also for homogeneous surfaces.

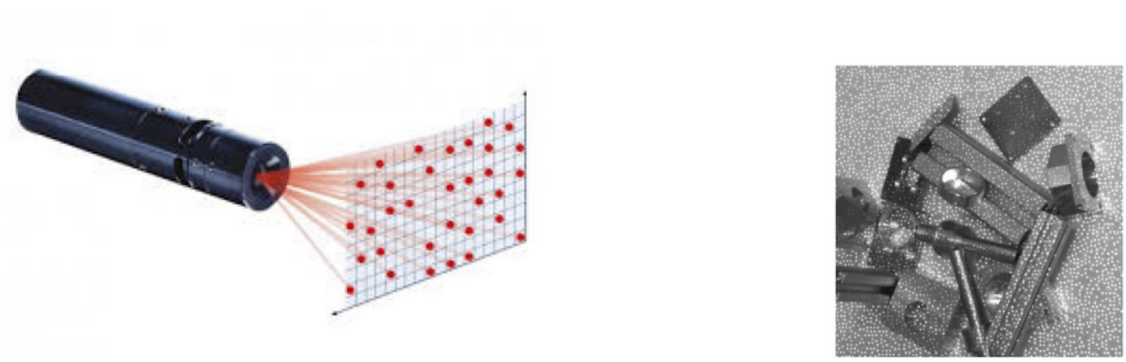


Figure 2.9: Pattern projector

Structured Light Cameras

They use the same principles of stereo cameras with the only difference that they use instead of two cameras, a single camera and a pattern projector which plays the role of a *virtual camera*. Briefly speaking, their working principle can be summarized as follows:

- Projection A light source (often a laser or LED) projects a structured pattern onto the object.
- Capture A camera records how the pattern distorts.
- Processing The system analyzes pattern distortions to calculate depth using triangulation.

Chapter 3

Kinematics of manipulators

In this chapter we will talk about *kinematics of manipulators*. How we will see, they are totally different with respect to mobile robots: if for mobile robots we have a moving chassis mounted on wheels, on the other hand we have chains of links and joints. This is due to substancial mechanical difference among the two cateogories. We will give an introduction on the structure of such robots (joint, links, wrists...) and then we will describe the *direct kinematics problem* and the *inverse kinematics problem*.

In this chapter we will see...

3.1 Kinematic chains

The **Kinematics** allows us to study the *position, velocity and acceleration* of particular points of a multibody system independently from forces and torques that generated them. In order to describe the *kinematic for manipulator* the definition of **kinematic chain** is needed.

Definition 3.1.1 (Kinematic chain)**.** A **kinematic chain (KC)** is a series of ideal arms/links connected by ideal joints.

For our purposes a KC is only a geometric entity, we will not consider mass, inertia, friction and so on. More specifically:

- *links/arms* are idealized with geometric bars connecting two or more joints;
- *Joints* are idealized physical components allowing a relative motion among consecutive arms. Each joint provides a **degree of motion**.

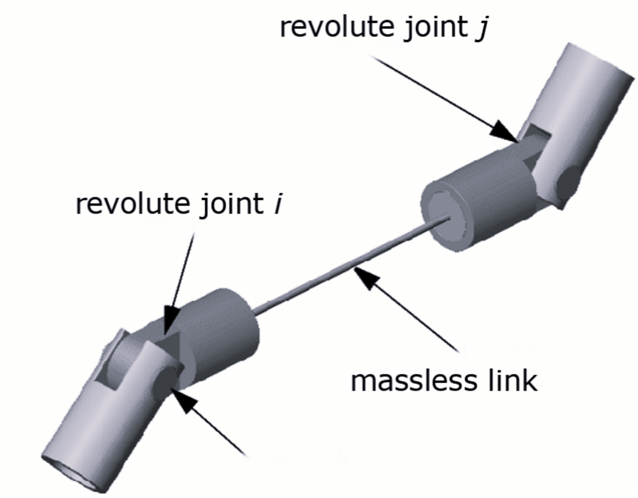


Figure 3.1: The robot joints are moved by actuators (eg. motors) and are connected by links/arm which are assumed to be massless

.

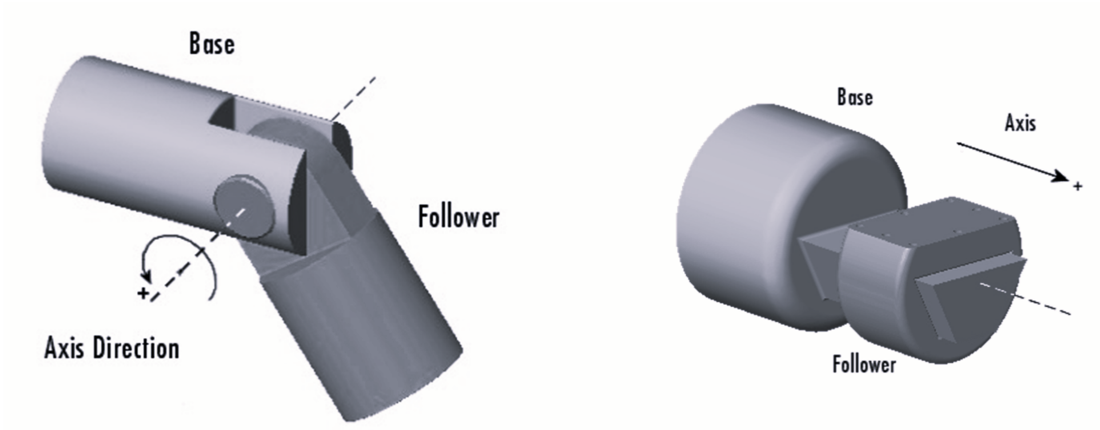


Figure 3.2: Revolute and Prismatic joints. With a dashed line the direction of motion (axis) is indicated

3.1.1 Types of joints

In the following we will consider two types of joints:

1. Revolute (or Rotational) joints which allows a **rotation** between the connected links;
2. Prismatic (or Traslational) joints which allows a **translation** between the connected links.

According to the number of links we can find between any two joints, kinematic chains can be **open chains** or **closed chains**. In the former case there is only one link between any two links in a way joints and arms form a *tree-like* structure; in the latter case there might be more than one link between any two joints, and in this case joints and arms are arranged in a *cycle-like* structure.

3.1.2 Graphical representation

There are different types of graphical representations for kinematic chains, in the following we will use cylinders and boxes for joints, segment for links/arms.

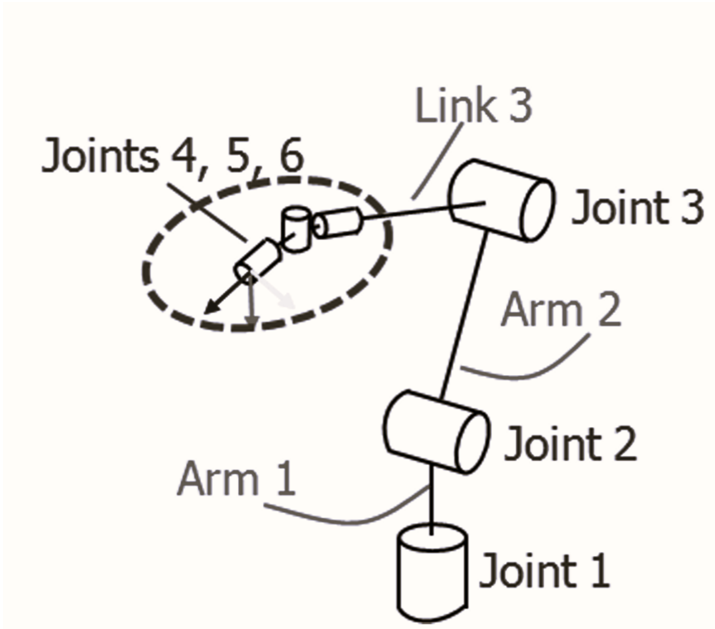


Figure 3.3: Kinematic chain sample

More specifically ***rotation joints*** are drawn in 3D has small cylinders withb the axes aligned along each rotation axis, in 2D they are drawn as small circles or small hourglasses.

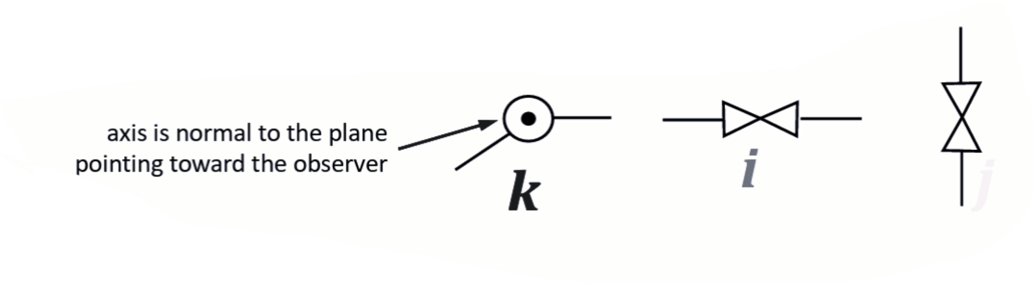


Figure 3.4: 2D-representation for revolute joints

At the opposite, ***prismatic joints*** are represented as small boex with each axis aligned along the translation axis. On the other hand in 2D they are drawn as small squares with a point in their center, or as small rectangles showing the direction for the outcoming links.

3.1.3 End-effector

The **end-effector** (also called *hand, gripper* or *hand tool*) is the structure which is attached to the last link, and it is the one by which the task, for which that robot was introduced, can be executed. For the end effector a particular type of point is interesting, this is the **Tool Center Point (TCP)** and it is the baricenter of the hand, or better that ideal point that the robot software moves through the space. How we will see it is useful to attach to such a point a reference frame.
The graphical representation is used for the end-effector is a sort of fork. The end-effector can be of any type and we can draw and study the kineamtic chain without assuming the use of any particular hand.

Figure 3.5: 2D representation of translational joints

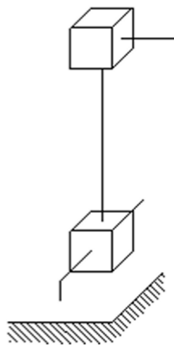


Figure 3.6: 3D graphical representation for prismatic joints

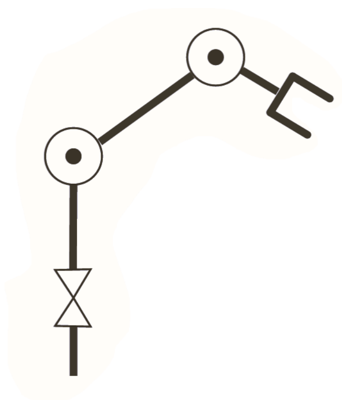


Figure 3.7: End-effector graphical representation. Typically the TCP is assumed to be the center of the 'fork'

3.2 Robot types

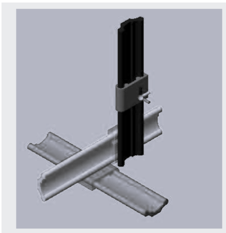
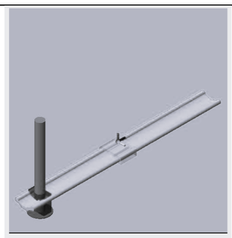
An **Industrial manipulator** is usually composed by a **shoulder** and a **wrist**. The manipulators can be categorized according to the structure of their arms, which are based on the type of joints. We will indicate:

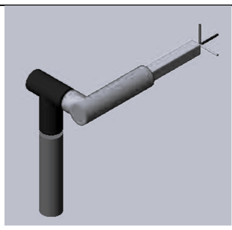
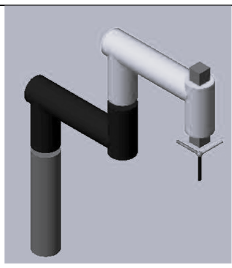
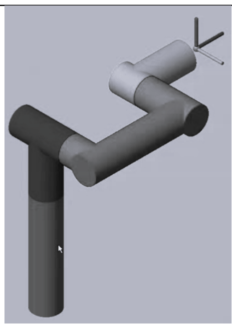
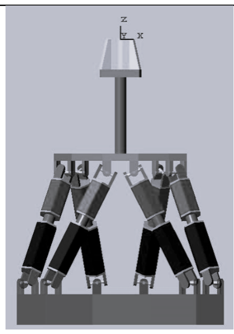
R=revolute joint
P=Prismatic joint

A great number of robots can be obtain according the shoulder configuration, however in the industrial fields, few configurations are used which are more suitable for certain tasks instead of others. In the following Table 3.1 we are going to explore the most common structures giving their main characteristics.

Table 3.1: Common types of manipulators

Table 3.1: Common types of manipulators

Robot type	Image	Description
Cartesian Manipulator		In a **cartesian manipulator** the structure `(PPP)` is related to the shoulder structure. In this case the shoulder is composed of three prismatic joints, whose axis are mutually orthogonal. In this case, each degree of motion corresponds to a cartesian variable. We anticipate that the task space is a sort of *parallelepiped*. Such robots have good accuracy, while regarding dexterity, it cannot be said the same.
Cylindrical Manipulator		One rotoidal joint and two prismatic joints are the basic building blocks for the shoulder of a cylindrical manipulator. Each DOM (degree of motion) corresponds to a cylindrical coordinate. The *task space* is a **cylindrical sector**. The horizontal prismatic joint allows reaching horizontal spaces; however, the accuracy decreases toward the arm ends.

Robot type	Image	Description
Spherical Manipulator		For a **polar (or spherical) manipulator**, the shoulder has two revolute joints followed by a prismatic one. Each DOM corresponds to a polar coordinate; here, the task is a spherical sector that may include parts of the floor to allow the manipulation of objects there located. The structure is less rigid compared to previous ones, and the accuracy reduces with the elongation of the prismatic arm.
SCARA (RRP)		It is a robot used for *pick and place* applications. The shoulder has two revolute joints followed by one prismatic joint, all with **parallel vertical axes**. The tasks addressed by this robot are the manipulation of small components or little assembly tasks.
Antropomorphic		The shoulder is composed of **three revolute joints**: the first one is vertical, the others are **horizontal and parallel**. It is one of the most common structures in the industry since it is the robot having **the best dexterity**.
Parallel robots		Such a type of manipulators have joints and links forming a cycle-like structure. They are not covered in these notes. An example is showed in the figure aside.

3.3 Wrists

The main scope of the **wrist** is to *give an orientation* to the TCP. In fact, it can be said that the shoulder sets the **origin position**, while the **wrist** orients the TCP. *Spherical wrists* are the most common. A wrist is said to be **spherical** if the three axes always intersect in a single point.Even if not spherical, due to its orientation function, the wrist is made up of **three rotational joints**.

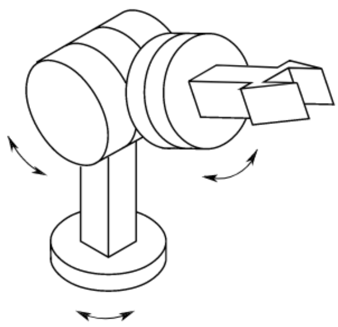


Figure 3.8: Spherical wrist

Taking into account a spherical joint, simplifies a lot the tractation of related topics of both manipulator kinematics and dynamics. Looking at the Figure 3.3 the wrist is the terminal part made up of joints 4,5,6.

3.4 Direct Kinematics

Many times in robotics there is the need to connect the "external world" to "robot world". In studying the kinematics of a manipulator the questions can arise are:

1. Where is the end-effector given the information about the joints? (This is known as the **Direct kinematics problem**).
2. What is the position of the joints to have in order to obtain a certain position of the end-effector? (This is known as the **Inverse kinematics problem**)

More specifically, the information about the joints are given by using **joint variables**, how we will see in explaining the Denavit-Hartemberg convention.

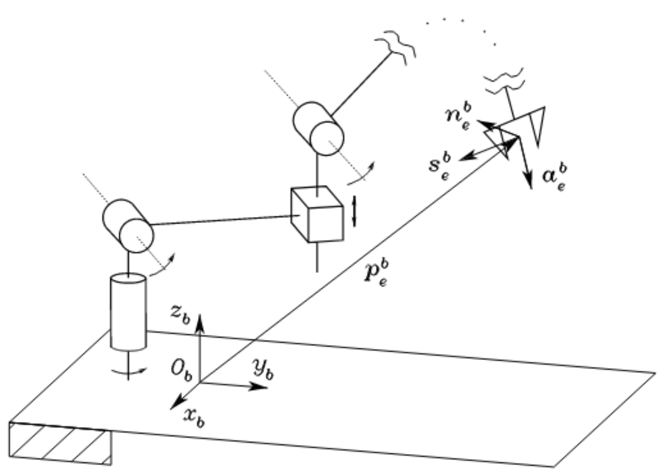


Figure 3.9: Kinematic chain: joints and arms

From now on, we are considering a manipulator consisting of $n+1$ links conected in an open chain through the use of n joints[1].
With respect to the reference frame attached to the base of the robot $\mathcal{R}_b = \{O, x_b, y_b, z_b\}$ the **direct kinematic function** is expressed in term of the homogeneuous transformation matrix

$$\boldsymbol{T}_e^b(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{n}_e^b(\boldsymbol{q}) & \boldsymbol{s}_e^b(\boldsymbol{q}) & \boldsymbol{a}_e^b(\boldsymbol{q}) & \boldsymbol{p}_e^b(\boldsymbol{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

where $\boldsymbol{q}$ is the vector containing the **joint variables** for the n joints. While $\mathcal{R}_e = \{O_e, n_e, s_e, a_e\}$ is a reference frame related to the end-effect (this is chosen according to the specific task geometry). Some further clarification about $\mathcal{R}_e$. If the end-effector is a gripper the origin of the frame is put in the TCP, the unit vectors are:

1. $\boldsymbol{a}$ stands for **approach**, lays along the approach direction;
2. $\boldsymbol{s}$ stands for **sliding** lays along the sliding plane of the gripper jaws.
3. $\boldsymbol{n}$ completes the right-handed frame.

A first approach to use to obtain the Equation (3.1) is by inspection and using geometrical considerations. A more efficient and relatively *effortless approach* is to use a systematic procedure. Things are made even more complicate when there are two or more closed kinematic chains. In the following we are going to briefly introduce the procedure for the case of *open kinematic chain.*

3.4.1 Open-chain manipulators and Denavit-Hartemberg convention

We have seen in Figure 3.9 that we have $n+1$ links with n number of joints. The Link 0 by convention is fixed to the ground, and (very important) **the actuation of the Link i moves link i**. Moreover, for each link we attach a reference frame so that when link i is actuated both the reference system and the link move.
Our objective is to relate the Link 0 with the Link n, looking for an homogeneous transformation between the frame n and the frame 0. This can be done in a recursive manner, computing for each link the relationship between the frame i and the frame $i-1$ by mean of matrices A_i^{i-1} where $i-1$ plays the role of fixed reference frame. At this point the *solution of the direct kinematic problem* is obtained by post-multiplying the homogeneous transformation matrices. That is

$$\boldsymbol{T}_n^0(\boldsymbol{q}) = A_1^0(q_1)A_2^1(q_2)\dots A_n^{n-1}(q_n) \tag{3.2}$$

Not rarely, the base and the end-effector frames does not coincide with the frames 0 and n, then the final $\boldsymbol{T}_e^b(\boldsymbol{q})$ is obtained as follows

$$\boldsymbol{T}_e^b(\boldsymbol{q}) = \boldsymbol{T}_0^b T_n^0(\boldsymbol{q}) T_e^n \tag{3.3}$$

The matrices relating the base with the Link 0 and the frame n with the end-effector frame are generally constant.
The **Denavit-Hartemberg convention** (DH convention) gives a systematic procedure to place the $n+1$ reference frames for each link and for obtaining the matrices A_i^{i-1} for each link.

$$A_i^{i-1}(q_i) = A_{i'}^{i-1} A_i^{i'} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

[1]It is sufficient to think that for a joint is for two links.

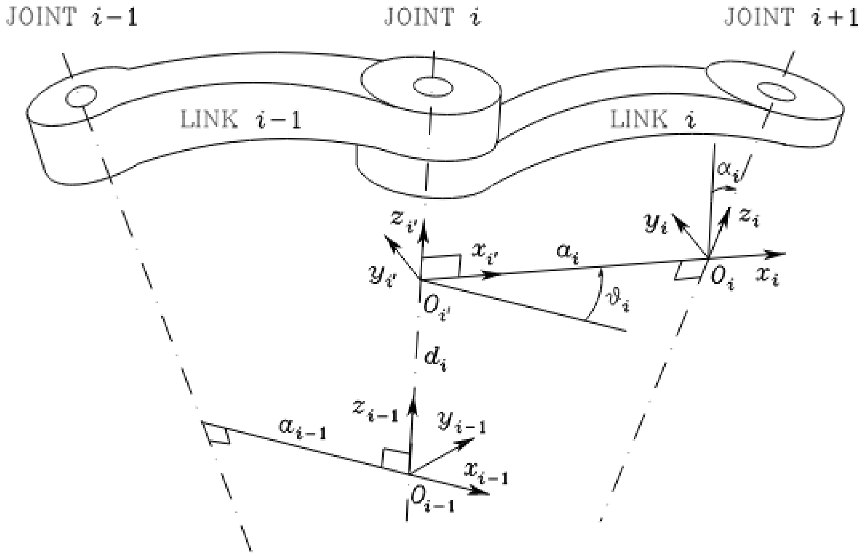


Figure 3.10: **DH parameters**. The z_i axis along the direction of motion for the related joint. O_i is the intersection of z_i with the common normal between z_{i-1} and z_i, while $O_{i'}$ is the intersection between the common normal and z_{i-1}. x_i, $x_{i'}$ are chosen along the common normal from $i-1$ to i. The unit vectors y_i, $y_{i'}$ are to complete the right-handed reference frames.

The parameters are summarized in the Figure 3.10 and by multiplying the matrices one can obtain the final homogeneous transformation matrix. Once the reference frames have been placed the DH parameters are defined as follows:

- a_i (*link length*) is the distance between origins O_i and $O_{i'}$;
- α_i (*link twist*) is the angle between z_{i-1} and z_i seen by the axis x_i (counterclockwise to be positive)
- θ_i (*joint angle*) is the angle between x_{i-1} and x_i seen by the axis z_{i-1} (counterclockwise to be taken positive)
- d_i (*link offset*) is the coordinate of $O_{i'}$ along the axis z_{i-1}

Remark. The Equation (3.4) has been obtained by (post)multipliying two successive homogeneous transformation: (i) from the frame $i-1$ to i'; (ii) from the frame i' to the frame i. This conceptual step of passing through the intermediate transformation can be skipped.

Remark. The reference frames to be placed are not univocally determined in the following cases:

- With respect to the 0-th reference frame in which only the direction of the z_0 is determined;
- With respect to the n-th reference frame since there is not a $n+1$ joint z_n is not uniquely determined. Typically, is chosen $z_n \parallel z_{n-1}$;
- when two consecutive axis are parallel since the common normal is not unique;
- when two axis intersect in a single point, the origin is unique, while x_i is arbitrary[2]
- when the i-th joint is prismatic only the direction of z_{i-1} is uniquely determined.

In such cases the indeterminedness can be exploited in order to simplify the procedure looking for alignment conditions between consecutive reference frames.

[2]Even if in order to measure an angle x_i must be orthogonal to both z_i and z_{i-1}

3.5 Other aspects about manipulators

3.5.1 Operational space and Joint space

We have seen till now that the *direct kinematics equation* allows to describe the position and orientation of the end-effector (the reference frame attached to it) in function of the joint variable with respect to the base reference frame. When we want to describe a task to be performed from the manipulator, we have the necessity to describe both position and orientation of $\mathcal{R}_\rceil$ in the time (*trajectories*).
As far as the position is concerned, we can proceed in a simple way while it is very difficult to guarantee the *orthogonality condition* during the time for the end-effector unit vectors $\boldsymbol{n}_e, \boldsymbol{s}_e, \boldsymbol{n}_e$. What is useful is to reduce the rotation matrix R_e^b to a minimal representation, for example the Euler angles associated to the rotation of $\mathcal{R}_e$ with respect to $\mathcal{R}_b$. In this way the end-effector pose can be expressed in terms of the following vector $\boldsymbol{x}_e$ having a number $m \leq 6$:

$$\boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{p}_e \\ \boldsymbol{\phi}_e \end{bmatrix} \tag{3.5}$$

where $\boldsymbol{p}_e$ is related to the position of O_e while ϕ_e expresses the orientation of the end-effector. Such an alternative representation is more confortable: since $\boldsymbol{k}$ is a vector function, tools from calculus (derivatives, gradients...) can be used. The space in which the vector x_e is defined is called **operational space** since we use it to describe the task to be performed by the manipulator. The space of the joint variables

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} \tag{3.6}$$

is called **joint space** (or **configuration space**). We recall that

$$q_i = \begin{cases} d_i & \text{Prismatic joint} \\ \theta_i & \text{Revolute joint} \end{cases} \tag{3.7}$$

At this point the direct kinematic equation can be also expressed in the following form

$$\boldsymbol{x}_e = \boldsymbol{k}(\boldsymbol{q}) \tag{3.8}$$

where the function $\boldsymbol{k}$ allows the computation of the operational space variables starting from the joint variables. Only in simple case finding an explicit form for $\boldsymbol{k}$ is straightforward.

3.5.2 Workspace

Taking into account the operational space, is defined **workspace** the set of points that the origin of R_e can assume when the joint assume all possible configurations. Often we use to distinguish between:

- *Reachable workspace* is the workspace that the end effector frame can describe with at least one orientation.
- *Dexterous workspace* is the workspace that $\mathcal{R}_e$ can reach by using many different configurations.

It appears clear that

$$\textsf{Reachable Workspace} \subseteq \textsf{Dexterous Workspace}$$

and points which are at the border of the workspace are reachable by uing a single orientation.

3.5.3 Accuracy and Repeteability

Accuracy

In a real manipulator there is always a discrepance between computed and real DH parameters, this is essentially due to mechanical tolerances. Since DH parameters are used in order to obtain the direct kinematics equation, the effective pose the manipulator attain with respect to the computed one by using the homogeneous transformation matrix $\boldsymbol{T}_e^b$. The *discrepancy* between real and computed solution is called **accuracy**. Modern manipulators guarantee an accuracy of the order of 1mm. In order to have a good accuracy, it is need a sufficiently rigid structure of the shoulder.

Repeteability

Refers to the ability of the manipulators to return to a previously reached position. It depends on the mechanical structure, on the sensors/transducers (especially from the resolution) and on the control strategies which are used, it is smaller than the accuracy. Modern manipulators have, even better performances with respect to the repeteability.

3.6 The *Inverse Kinematics* problem

The **inverse kinematics problem** is related to finding what is $\boldsymbol{q}$ (joint variables) which gives a certain end-effector position and orientation $\boldsymbol{x}_e$. This is not a simple problem since: (i) closed form solutions could not exist; (ii) multiple or infinite solutions could exist; (iii) there are cases in which no solutions are available. There is a nice property which guarantee the existence of a solution if $\boldsymbol{x}_e$ belongs to the dexterous workspace.
The *inverse kinmeatic problem* could be solved by using algebraic or geometrical intuitions. In the former case we have to solve nonlinear trascendent equations, in the latter case we are looking for significant points which lead to the provided position and orientation of the end-effector. Very often **numerical methods** are used in order to find a valid configuration for which a certain pose of the end-effector is attained.

$$\text{given} \quad \boldsymbol{x}_e(\boldsymbol{q}) \rightarrow \text{I want } \boldsymbol{q} \tag{3.9}$$

In the next chapter about *differential kinematics* we will find a way for inverting the kinematic equation using the **jacobian of the manipulator**.

3.6.1 (Analytic) Solution for manipulators with spherical wrist

The great majority of existing manipulators are simple from a kinematic point of view, since they are composed of a shoulder whose objective is to give a position to O_e with typical structures (cylindrical, polar, antropomorphic...) and a spherical wrist for the orientation of $\mathcal{R}_e$. This is not a casual choice, since it derives from the difficulty in finding the solution for the inverse kinematic problem.
It holds that for 6dof manipulators there is an analytic solution to the problem if:

1. Three axis of adjacent rotoidal joints intersect in a point (case of spherical wrist);
2. Three axis of adjacent rotoidal joints are parallel the one with respect to the others.

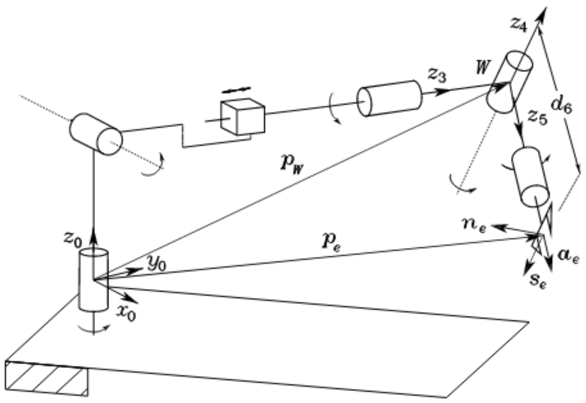


Figure 3.11: Example of manipulator with spherical wrist

When this is possible we can articulate the solution for the inverse kinematics in two parts. In particular given the position and orientation of the end effector $\boldsymbol{p}_e$ and $\boldsymbol{R}_e$: (i) we find the joint variables of the shoulder inverting the $\boldsymbol{p}_e(q_1, q_2, q_3)$, then using $\boldsymbol{R}_0^3$ and $\boldsymbol{R}_e$ we find the direct kinematics relation for the wrist whose inversion leads to the remaining joint variables $\theta_1, \theta_2, \theta_3$ which gives the orientation of the end-effector.
In this way we are solving separetely the inverse kinematics for the shoulder and for the spherical wrist.

Chapter 4

Differential Kinematics of manipulators

4.1 Geometric Jacobian

4.2 Kinematic singularities