

Modeling and Control of Manipulators

Part I: Geometric and Kinematic Models

Wisama KHALIL

Ecole Centrale de Nantes

Master I EMARO

European Master on Advanced Robotics

Master I ARIA

**Master Automatique, Robotique et Informatique
Appliquée**

**Master in Control Engineering, Robotics and Applied
Informatics**

2013-2014



Education and Culture

Erasmus Mundus



*Centrale
Nantes*

Table of Contents

Chapter 1: Terminology and general definitions

1.1. Introduction	1
1.2. Mechanical components of a robot	2
1.3. Definitions	4
1.3.1. Joints	4
1.3.2 Degrees of Mobility or Number of degrees of freedom of a body	5
1.3.3. Joint space	5
1.3.4. Task space	5
1.3.5. Redundancy	6
1.3.6. Singular configurations	6
1.4. Choosing the number of degrees of freedom of a robot	7
1.5. Architectures of robot manipulators	7
1.6. Characteristics of a robot	12
1.7. Conclusion	13

Chapter 2: Transformation matrix between vectors, frames and screws

Transformation matrix between vectors, frames and screws

2.1. Introduction	15
2.2. Homogeneous coordinates	16
2.2.1. Representation of a point	16
2.2.2. Representation of a direction.....	16
2.2.3. Representation of a plane	17
2.3. Homogeneous transformations [Paul 81].....	17
2.3.1. Transformation of frames	17
2.3.2. Transformation of vectors	18
2.3.3. Transformation of planes	19
2.3.4. Transformation matrix of a pure translation	19
2.3.5. Transformation matrices of a rotation about the principle axes	20
2.3.6. Properties of homogeneous transformation matrices	22
2.3.7. Transformation matrix of a rotation about a general vector located at the origin .	25
2.3.8. Equivalent angle and axis of a general rotation	27
2.4. Representation of velocities	29
2.4.1. Definition of a screw	30
2.4.2. Kinematic screw	30
2.4.3. Transformation matrices between screws	31
2.5. Differential translation and rotation of frames	32
2.6. Representation of forces (wrench)	35
2.7. Conclusion	36

Chapter 3: Direct geometric model of serial robots

3.1. Introduction	37
3.2. Description of the geometry of serial robots	38
3.3. Direct geometric model	44
3.4. Optimization of the computation of the direct geometric model	47
3.5. Transformation matrix of the end-effector in the world frame	50
3.6. Specification of the orientation	51
3.6.1. Euler angles	52
3.6.2. Roll-Pitch-Yaw angles	54
3.6.3. Quaternions	56
3.7. Conclusion	58

Chapter 4: Inverse geometric model of serial robots

4.1. Introduction	59
4.2. Mathematical statement of the problem	60
4.3. Inverse geometric model of robots with simple geometry	61
4.3.1. Principle	61
4.3.2. Special case: robots with a spherical wrist	63
4.3.3. Inverse geometric model of robots with more than six degrees of freedom	71
4.3.4. Inverse geometric model of robots with less than six degrees of freedom	71
4.4. Inverse geometric model of decoupled six degree-of-freedom robots	74
4.4.1. Introduction	74
4.4.2. Inverse geometric model of six degree-of-freedom robots with a spherical joint	75
4.4.3. Inverse geometric model of robots with three prismatic joints	82
4.5. Inverse geometric model of general 6 dof robots	83
4.6. Conclusion	87

Chapter 5: Direct kinematic model of serial robots

5.1. Introduction	89
5.2. Computation of the Jacobian matrix from the direct geometric model	90
5.3. Kinematic Jacobian matrix	91
5.3.1. Computation of the kinematic Jacobian matrix	92
5.3.2. Computation of the matrix iJ_n	94
5.4. Decomposition of the Jacobian matrix into three matrices	97
5.5. Efficient computation of the end-effector velocity	98
5.6. Dimension of the task space of a robot	99
5.7. Analysis of the robot workspace	100
5.7.1. Workspace	100
5.7.2. Singularity branches	101
5.7.3. Jacobian surfaces	104
5.7.4. Concept of aspect	105
5.7.5. t-connected subspaces	107
5.8. Velocity transmission between joint space and task space	110
5.8.1. Singular value decomposition	110
5.8.2. Velocity ellipsoid: velocity transmission performance	112

5.9. Static model	114
5.9.1. Representation of a wrench	114
5.9.2. Mapping of an external wrench into joint torques	114
5.9.3. Velocity-force duality	115
5.10. Second order kinematic model	117
5.11. Kinematic model associated with the task coordinates representation ..	118
5.11.1. Direction cosines	119
5.11.2. Euler angles	120
5.11.3. Roll-Pitch-Yaw angles	121
5.11.4. Quaternions	122
5.12. Conclusion	122
 Chapter 6: Inverse kinematic model of serial robots	
6.1. Introduction	124
6.2. General form of the kinematic model	124
6.3. Inverse kinematic model for a regular case	125
6.3.1. First method	126
6.3.2. Second method	126
6.4. Solution in the neighborhood of singularities	128
6.4.1. Use of the pseudoinverse	129
6.4.2. Use of the damped pseudoinverse	130
6.4.3. Other approaches for controlling motion near singularities	132
6.5. Inverse kinematic model of redundant robots	133
6.5.1. Extended Jacobian	133
6.5.2. Use of the pseudoinverse of the Jacobian matrix	135
6.5.3. Weighted pseudoinverse	135
6.5.4. Pseudoinverse solution with an optimization term	136
6.5.5. Task-priority concept	138
6.6. Numerical calculation of the inverse geometric problem	141
6.7. Minimum description of tasks [Fournier 80], [Dombre 81]	142
6.7.1. Principle of the description	143
6.7.2. Differential models associated with the minimum description of Tasks	146
6.8. Conclusion	153
 References	467
Appendix 1: Solution of the inverse geometric model equations (Table 4.1).....	497
Appendix 2: The inverse robot.....	503
Appendix 3: Dyalitic elimination	505
Appendix 4: Solution of systems of linear equations.....	507

Modélisation, identification et commande des robots

Chapter 1

Terminology and general definitions

1.1. Introduction

A robot is a mechanical device, containing electrically, electronically and IT parts. It possesses capacities of perception, action, decision, learning, communication and interaction with its environment, to realize certain tasks on the place of the man or in interaction with the man. Robotics is the field concerned with designing, constructing, integrating and programming robots.

Robots have been widely used with success in various industrial applications. Since the last two decades, other areas of application have emerged: medical, service (spatial, civil security, ...), transport, underwater, entertainment, and even providing companionship in the form of artificial “pets” or humanoid robots. We can distinguish three main classes of robots: *robot manipulators*, which imitate the human arm, *walking robots*, which imitate the locomotion of humans, animals or insects, *mobile robots*, which look like cars, and flying robots “drones”.

The terms *adaptability* and *versatility* are often used to highlight the intrinsic flexibility of a robot. Adaptability means that the robot is capable of adjusting its motion to comply with environmental changes during the execution of tasks. Versatility means that the robot may carry out a variety of tasks – or the same task in different ways – without changing the mechanical structure or the control system.

A robot is composed of the following subsystems:

- *mechanism*: consists of an articulated mechanical structure actuated by electric, pneumatic or hydraulic actuators, which transmit their motion to the joints using suitable transmission systems;
- *perception capabilities*: They consist of the internal sensors that provide information about the state of the robot (joint positions and velocities), and the external sensors to obtain the information about the environment (contact

2 Modeling, identification and control of robots

detection, distance measurement, artificial vision). They help the robot to adapt to disturbances and unpredictable changes in its environment;

- *controller*: realizes the desired task objectives. It generates the input signals for the actuators as a function of the user's instructions and the sensor outputs;
- *communication interface*: through this the user programs the tasks that the robot must carry out;
- *workcell and peripheral devices*: constitute the environment in which the robot works.

Robotics is thus a multidisciplinary science, which requires a background in mechanics, automatic control, electronics, signal processing, communications, computer engineering, etc.

The objective of this book is to present the techniques of the modeling, identification and control of robots. We restrict our study to rigid robot manipulators with a fixed base.

In this chapter, we will present certain definitions that are necessary to classify the mechanical structures and the characteristics of robot manipulators.

1.2. Mechanical components of a robot

The mechanism of a robot manipulator consists of two distinct subsystems, one (or more) end-effectors and an articulated mechanical structure:

- by the term *end-effector*, we mean any device intended to manipulate objects (magnetic, electric or pneumatic grippers) or to transform them (tools, welding torches, paint guns, etc.). It constitutes the interface with which the robot interacts with its environment. An end-effector may be multipurpose, i.e. equipped with several devices each having different functions;
- the role of *the articulated mechanical structure* is to place the end-effector at a given location (position and orientation) with a desired velocity and acceleration. The mechanical structure is composed of a kinematic chain of articulated rigid links. One end of the chain is fixed and is called the *base*. The end-effector is fixed to the free extremity of the chain. This chain may be *serial (simple open chain)* (Figure 1.1), *tree structured* (Figure 1.2) or *closed* (Figures 1.3 and 1.4). The last two structures are termed *complex chains* since they contain at least one link with more than two joints.

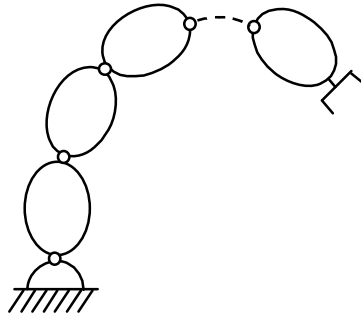


Figure 1.1. *Simple open (or serial) chain*

Serial robots with a simple open chain are the most commonly used. There are also industrial robots with closed kinematic chains, which have the advantage of being more rigid and accurate.

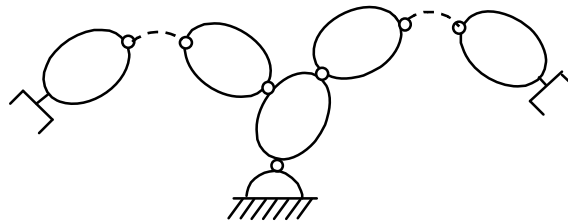


Figure 1.2. *Tree structured chain*

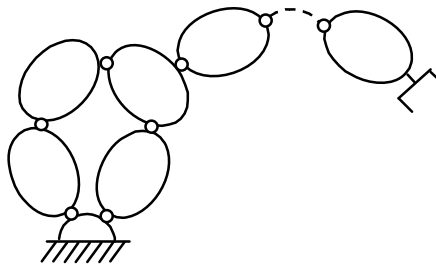


Figure 1.3. *Closed chain*

Figure 1.4 shows a specific architecture with closed chains, which is known as a *parallel robot*. In this case, the end-effector is connected to the base by several parallel chains [Inoue 85], [Fichter 86], [Reboulet 88], [Gosselin 88], [Clavel 89],

[Charentus 90], [Pierrot 91a], [Merlet 00]. The mass ratio of the payload to the robot is much higher compared to serial robots. This structure seems promising in manipulating heavy loads with high accelerations and realizing difficult assembly tasks.

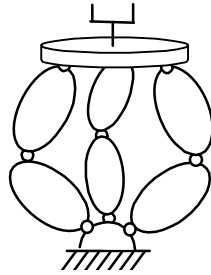


Figure 1.4. Parallel robot

1.3. Definitions

1.3.1. Joints

A joint connects two successive links, thus limiting the number of degrees of freedom between them. The resulting number of degrees of freedom, m , is also called *joint mobility*, such that $0 \leq m \leq 6$.

When $m = 1$, which is frequently the case in robotics, the joint is either *revolute* or *prismatic*. A complex joint with several degrees of freedom can be constructed by an equivalent combination of revolute and prismatic joints. For example, a spherical joint can be obtained by using three revolute joints whose axes intersect at a point.

1.3.1.1. Revolute joint

This limits the motion between two links to a rotation about a common axis. The relative location between the two links is given by the angle about this axis. The revolute joint, denoted by R , is represented by the symbols shown in Figure 1.5.

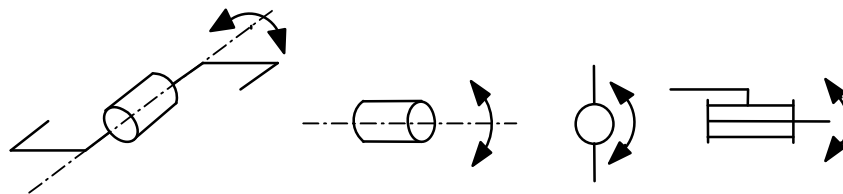


Figure 1.5. Symbols of a revolute joint

1.3.1.2. Prismatic joint

This limits the motion between two links to a translation along a common axis. The relative location between the two links is determined by the distance along this axis. The prismatic joint, denoted by P, is represented by the symbols shown in Figure 1.6.

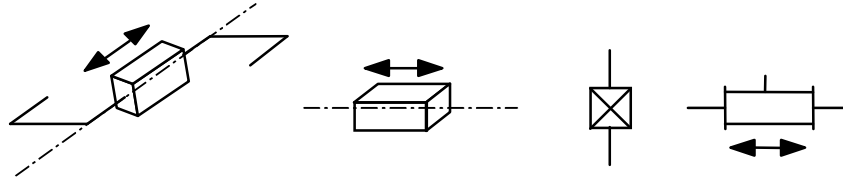


Figure 1.6. Symbols of a prismatic joint

1.3.2 Mobility or Number of degrees of freedom of a body

The mobility of a body is defined as the number of independent components of its instantaneous velocity (rotational and linear), thus it is equal to 6 for a free body in space and 3 for a body in plane. In general it is ≤ 6 .

1.3.3. Joint space $E(q)$

The space in which the location of all the links of a robot are represented is called *joint space*, or *configuration space*. We use the *joint variables*, $\mathbf{q} \in \mathbb{R}^N$, as the coordinates of this space. Its dimension N is equal to the number of independent joints and corresponds to the number of degrees of freedom of the mechanical structure also known as the mobility of the structure. In an open chain robot (simple or tree structured), the joint variables are generally independent, whereas a closed chain structure implies constraint relations between the joint variables.

Unless otherwise stated, we will consider that a robot with N degrees of freedom has N actuated joints.

1.3.4. Task space $E(X)$

The location, position and orientation, of the end-effector is represented in the *task space*, or *operational space*. We may consider as many task spaces as there are end-effectors. Generally, Cartesian coordinates are used to specify the position in \mathbb{R}^3 and the rotation group $SO(3)$ for the orientation. Thus the task space is equal to

$\mathbb{R}^3 \times \text{SO}(3)$. An element of the task space is represented by the vector $\mathbf{X} \in \mathbb{R}^M$, where M is equal to the maximum number of independent parameters that are necessary to specify the location of the end-effector in space. Consequently, $M \leq 6$ and $M \leq N$. In robot-manipulator (where holonomic joints are used) M is equal to the maximum mobility of the end-effector.

1.3.5. Redundancy

A robot is classified as *redundant* when the number of degrees of freedom of its joint space is greater than the number of degrees of freedom of its task space ($N > M$). Such robot can have an infinite number of configurations to locate the end effector at a desired location. This property increases the volume of the reachable workspace of the robot and enhances its performance. We will see in Chapter 6 that redundant robots can achieve a secondary optimum objective besides the primary objective of locating the end-effector.

Notice that a simple open chain is redundant if it contains any of the following combinations of joints (non-exhaustive):

- more than six joints;
- more than three revolute joints whose axes intersect at a point;
- more than three revolute joints with parallel axes;
- more than three prismatic joints;
- prismatic joints with parallel axes;
- revolute joints with collinear axes.

NOTES.–

- for an articulated mechanism with several end-effectors, redundancy is evaluated by comparing the number of degrees of freedom of the joint space acting on each end-effector and the number of degrees of freedom of the corresponding task space;
- a robot which is not redundant ($N=M$) may be redundant with respect to a particular task whose number of degrees of freedom, m , is less than the number of degrees of freedom of the robot.
- A robot is said to have redundant actuators if the number of motorized joints is greater than the number of degrees of freedom of the robot. This case may take place only in closed loop structures.

1.3.6. Singular configurations

For all robots, redundant or not, it is possible that at some configurations, called *singular configurations*, the number of degrees of freedom of the end-effector becomes less than the dimension of the task space. For example, this may occur when:

- the axes of two prismatic joints become parallel;
- the axes of two revolute joints become collinear;

In Chapter 5, we will present a mathematical condition to determine the number of degrees of freedom of the task space of a mechanism as well as its singular configurations.

1.4. Choosing the number of degrees of freedom of a robot

A non-redundant robot must have six degrees of freedom in order to place an arbitrary object in space. However, if the manipulated object exhibits revolution symmetry, five degrees of freedom are sufficient, since it is not necessary to specify the rotation about the revolution axis. In the same way, to locate a body in a plane, one needs only three degrees of freedom: two for positioning a point in the plane and the third to determine the orientation of the body.

From these observations, we deduce that:

- the number of degrees of freedom of a mechanism is chosen as a function of the shape of the object to be manipulated by the robot and of the class of tasks to be realized;
- a necessary but insufficient condition to have compatibility between the robot and the task is that the number of degrees of freedom of the end-effector of the robot is equal to or more than that of the task.

1.5. Architectures of robot manipulators

Without anticipating the results of the next chapters, we can say that the study of both tree structured and closed chains can be reduced to some equivalent simple open chains. Thus, the classification presented below is relevant for simple open chain architectures, but may also be generalized to the complex chains.

In order to count the possible architectures, we only consider revolute or prismatic joints whose consecutive axes are either parallel or perpendicular. Generally, with some exceptions (in particular, the last three joints of the GMF P150 and Kuka IR600 robots), the consecutive axes of currently used robots are either parallel or perpendicular. The different combinations of these four parameters yield the number of possible architectures with respect to the number of joints as shown in Table 1.1 [Delignières 87], [Chedmail 90a].

The first three joints of a robot are commonly designed in order to perform gross motion of the end-effector, and the remaining joints are used to accomplish orientation. Thus, the first three joints and the associated links constitute the shoulder or regional positioning structure. The other joints and links form the wrist.

Taking into account these considerations and the data of Table 1.1, one can count 36 possible combinations of the shoulder. Among these architectures, only 12 are mathematically distinct and non-redundant (we eliminate, *a priori*, the structures limiting the motion of the terminal point of the shoulder to linear or planar displacement, such as those having three prismatic joints with parallel axes, or three revolute joints with parallel axes). These structures are shown in Figure 1.7.

Table 1.1. Number of possible architectures as a function of the number of degrees of freedom of the robot

Number of degrees of freedom of the robot	Number of architectures
2	8
3	36
4	168
5	776
6	3508

A survey of industrial robots has shown that only the following five structures [Liégeois 79] are manufactured:

- anthropomorphic shoulder represented by the first RRR structure shown in Figure 1.7, like PUMA from Unimation, Acma SR400, ABB IRBx400, Comau Smart-3, Fanuc (S-xxx, Arc Mate), Kuka (KR 6 to KR 200), Reis (RV family), Staübli (RX series), etc.;
- spherical shoulder RRP: "Stanford manipulator" and Unimation robots (Series 1000, 2000, 4000);
- RPR shoulder corresponding to the first RPR structure shown in Figure 1.7: Acma-H80, Reis (RH family), etc. The association of a wrist with one revolute degree of freedom of rotation to such a shoulder can be found frequently in the industry. The resulting structure of such a robot is called SCARA (Selective Compliance Assembly Robot Arm) (Figure 1.8). It has several applications, particularly in planar assembly. SCARA, designed by Sankyo, has been manufactured by many other companies: IBM, Bosch, Adept, etc.;

- cylindrical shoulder RPP: Acma-TH8, AFMA (ROV, ROH), etc.;
- Cartesian shoulder PPP: Acma-P80, IBM-7565, Sormel-Cadratic, Olivetti-SIGMA. More recent examples: AFMA (RP, ROP series), Comau P-Mast, Reis (RL family), SEPRO, etc.

The second RRR structure of Figure 1.7, which is equivalent to a spherical joint, is generally used as a wrist. Other types of wrists are shown in Figure 1.9 [Delignières 87].

A robot, composed of a shoulder with three degrees of freedom and a spherical wrist, constitutes a classical six degree-of-freedom structure (Figure 1.10). Note that the position of the center of the spherical joint depends only on the configuration of joints 1, 2 and 3. We will see in Chapter 4 that, due to this property, the inverse geometric model, providing the joint variables for a given location of the end-effector, can be obtained analytically for such robots.

According to the survey carried out by the French Association of Industrial Robotics (AFRI) and RobAut Journal [Fages 98], the classification of robots in France (17794 robots), with respect to the number of degrees of freedom, is as follows: 4.5% of the robots have three degrees of freedom, 27% have four, 9% have five and 59.5% have six or more. As far as the architecture of the shoulder is concerned, there is a clear dominance of the RRR anthropomorphic shoulder (65.5%), followed by the Cartesian shoulder (20.5%), then the cylindrical shoulder (7%) and finally the SCARA shoulder (7%).

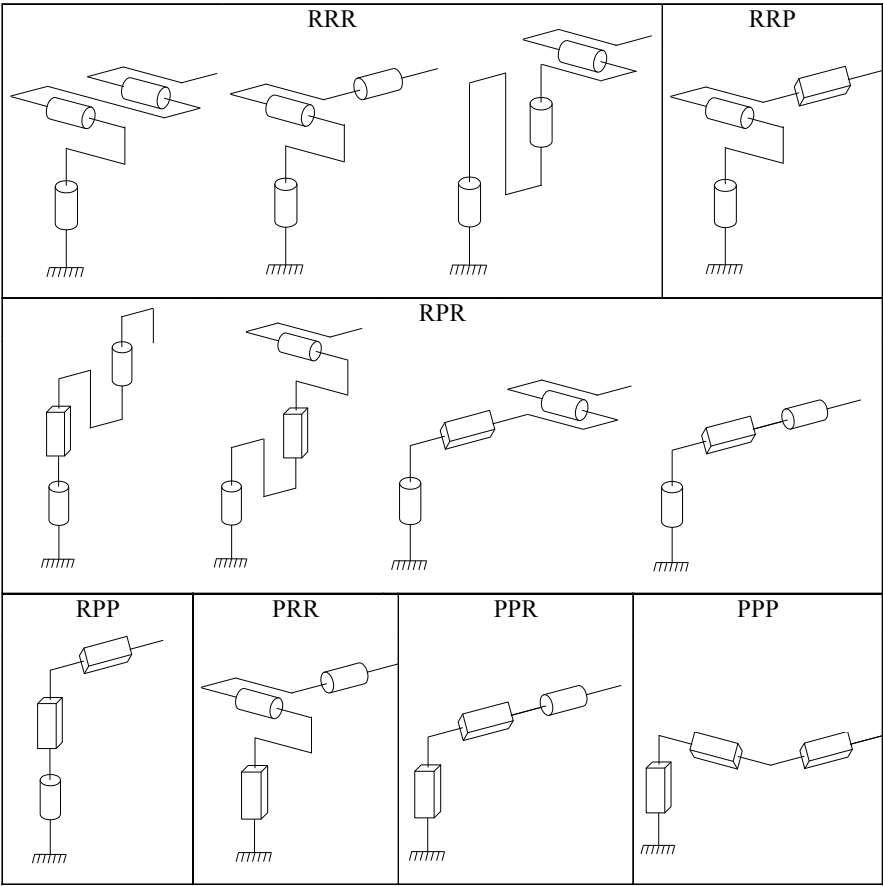


Figure 1.7. Architectures of the shoulder (from [Milenkovic 83])

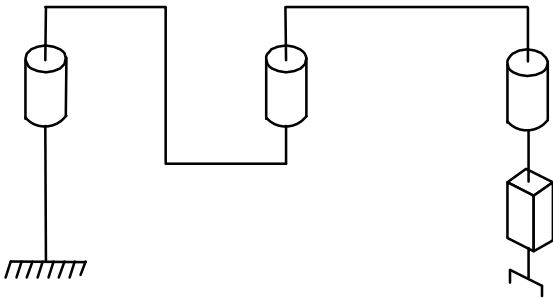


Figure 1.8. SCARA robot

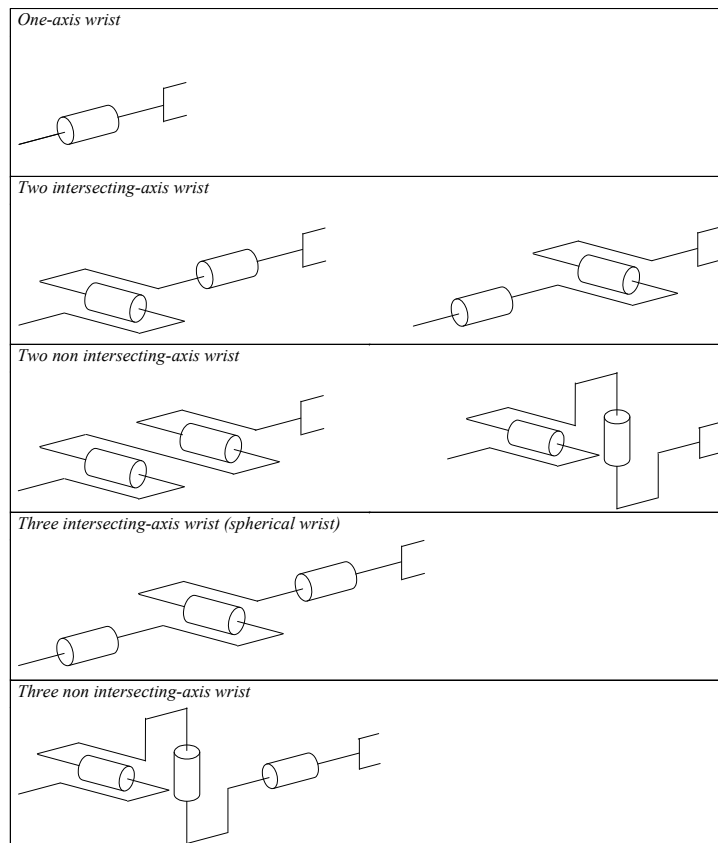


Figure 1.9. Architectures of the wrist (from [Delignières 87])

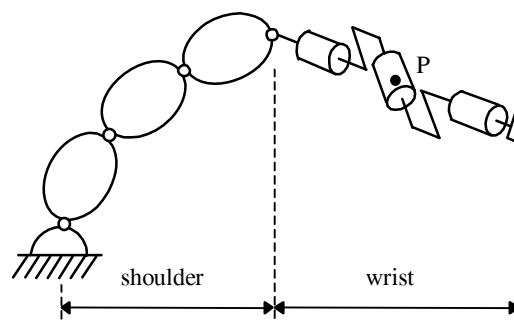


Figure 1.10. Classical six degree-of-freedom robot

1.6. Characteristics of a robot

The standard ISO 9946 specifies the characteristics that manufacturers of robots must provide. Here, we describe some of these characteristics that may help the user in choosing an appropriate robot with respect to a given application:

- *workspace*: defines the space that can be reached by the end-effector. Its range depends on the number of degrees of freedom, the joint limits and the length of the links;
- *payload*: maximum load carried by the robot;
- *maximum velocity and acceleration*: determine the cycle time;
- *position accuracy* (Figure 1.11): indicates the difference between a commanded position and the mean of the attained positions when visiting the commanded position several times from different initial positions;
- *position repeatability* (Figure 1.11): specifies the precision with which the robot returns to a commanded position. It is given as the distance between the mean of the attained positions and the furthestmost attained position;
- *resolution*: the smallest increment of movement that can be achieved by the joint or the end-effector.

However, other characteristics must also be taken into account: technical (energy, control, programming, etc.) and commercial (price, maintenance, etc.). Thus, the selection criteria are sometimes difficult to formulate and are often contradictory. To a certain extent, the simulation and modeling tools available in Computer Aided Design (CAD) packages may help in making the best choice [Dombre 88b], [Zeghloul 91], [Chedmail 92], [Chedmail 98].

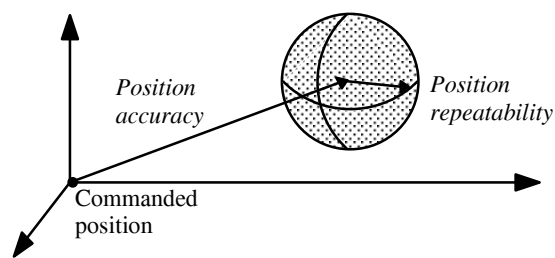


Figure 1.11. Position accuracy and repeatability (from [Priel 90])

1.7. Conclusion

In this chapter, we have presented the definitions of some technical terms related to the field of modeling, identification and control of robots. We will frequently come across these terms in this book and some of them will be reformulated in a more analytical or mathematical way. The figures mentioned here justify the choice of the robots that are taken as examples in the following chapters. In the next chapter, we present the transformation matrix concept, which constitutes an important mathematical tool for the modeling of robots.

Chapter 2

Transformation matrix between vectors, frames and screws

2.1. Introduction

In robotics, we assign one or more frames to each link of the robot and each object of the workcell. Thus, transformation of frames is a fundamental concept in the modeling and programming of a robot. It enables us to:

- compute the location, position and orientation of robot links relative to each other;
- describe the position and orientation of objects;
- specify the trajectory and velocity of the end-effector of a robot for a desired task;
- describe and control the forces when the robot interacts with its environment;
- implement sensory-based control using information provided by various sensors, each having its own reference frame.

In this chapter, we present a notation that allows us to describe the relationship between different frames and objects of a robotic cell. This notation, called *homogeneous transformation*, has been widely used in computer graphics [Roberts 65], [Newman 79] to compute the projections and perspective transformations of an object on a screen. Currently, this is also being used extensively in robotics [Pieper 68], [Paul 81]. We will show how the points, vectors and transformations between frames can be represented using this approach. Then, we will define the differential transformations between frames as well as the representation of velocities and forces using screws.

2.2. Homogeneous coordinates

2.2.1. Representation of a point

Let (iP_x, iP_y, iP_z) be the Cartesian coordinates of an arbitrary point P with respect to the frame R_i , which is described by the origin O_i and the axes x_i, y_i, z_i (Figure 2.1). The homogeneous coordinates of P with respect to frame R_i are defined by $(w^iP_x, w^iP_y, w^iP_z, w)$, where w is a scaling factor. In robotics, w is taken to be equal to 1. Thus, we represent the homogeneous coordinates of P by the (4×1) column vector:

$${}^i\mathbf{P} = {}^i(\mathbf{O}_i \mathbf{P}) = \begin{bmatrix} iP_x \\ iP_y \\ iP_z \\ 1 \end{bmatrix} \quad [2.1]$$

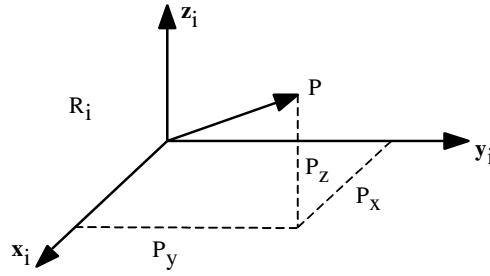


Figure 2.1. Representation of a point vector

2.2.2. Representation of a direction

A direction (free vector) is also represented by four components, but the fourth component is zero, indicating a vector at infinity. If the Cartesian coordinates of a unit vector \mathbf{u} with respect to frame R_i are $(i u_x, i u_y, i u_z)$, its homogeneous coordinates will be:

$${}^i\mathbf{u} = \begin{bmatrix} i u_x \\ i u_y \\ i u_z \\ 0 \end{bmatrix} \quad [2.2]$$

2.2.3. Representation of a plane

The homogeneous coordinates of a plane Q , whose equation with respect to a frame R_i is ${}^i\alpha x + {}^i\beta y + {}^i\gamma z + {}^i\delta = 0$, are given by:

$${}^iQ = [{}^i\alpha \quad {}^i\beta \quad {}^i\gamma \quad {}^i\delta] \quad [2.3]$$

If a point P lies in the plane Q , then the matrix product ${}^iQ {}^iP$ is zero:

$${}^iQ {}^iP = [{}^i\alpha \quad {}^i\beta \quad {}^i\gamma \quad {}^i\delta] \begin{bmatrix} {}^iP_x \\ {}^iP_y \\ {}^iP_z \\ 1 \end{bmatrix} = 0 \quad [2.4]$$

2.3. Homogeneous transformations [Paul 81]

2.3.1. Transformation of frames

The transformation, translation and/or rotation, of a frame R_i into frame R_j (Figure 2.2) is represented by the (4x4) homogeneous transformation matrix iT_j such that:

$${}^iT_j = \begin{bmatrix} {}^i s_j & {}^i n_j & {}^i a_j & {}^i p_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.5a]$$

where ${}^i s_j$, ${}^i n_j$ and ${}^i a_j$ contain the components of the unit vectors along the x_j , y_j and z_j axes respectively expressed in frame R_i , and where ${}^i p_j$ is the vector representing the coordinates of the origin of frame R_j expressed in frame R_i .

We can also say that the matrix iT_j defines frame R_j relative to frame R_i . Thereafter, the transformation matrix [2.5a] will occasionally be written in the form of a partitioned matrix:

$${}^iT_j = \begin{bmatrix} {}^iR_j & {}^iP_j \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i s_j & {}^i n_j & {}^i a_j & {}^i p_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.5b]$$

Apparently, this is in violation of the homogeneous notation since the vectors have only three components. In any case, the distinction in the representation with either three or four components will always be clear in the text.

In summary:

- the matrix ${}^i\mathbf{T}_j$ represents the transformation from frame R_i to frame R_j ;
- the matrix ${}^i\mathbf{T}_j$ can be interpreted as representing the frame R_j (three orthogonal axes and an origin) with respect to frame R_i .

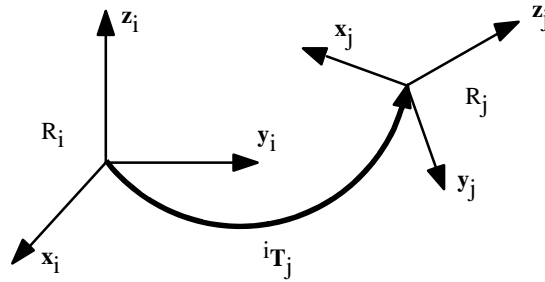


Figure 2.2. Transformation of frames

2.3.2. Transformation of vectors

Let the vector ${}^j\mathbf{P}$ define the homogeneous coordinates of the point P with respect to frame R_j (Figure 2.3). Thus, the homogeneous coordinates of P with respect to frame R_i can be obtained as:

$${}^i\mathbf{P} = {}^i(\mathbf{O}_i\mathbf{P}) = {}^i\mathbf{s}_j {}^jP_x + {}^i\mathbf{n}_j {}^jP_y + {}^i\mathbf{a}_j {}^jP_z + {}^i\mathbf{P}_j = {}^i\mathbf{T}_j {}^j\mathbf{P} \quad [2.6]$$

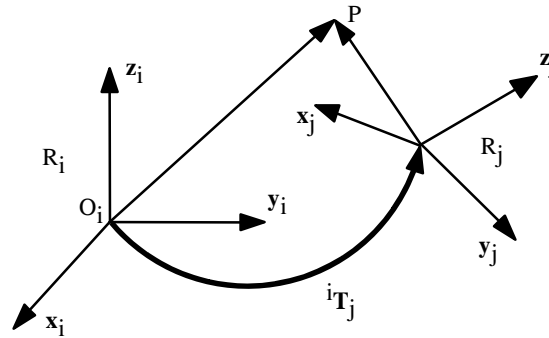


Figure 2.3. Transformation of a vector

Thus the matrix ${}^i\mathbf{T}_j$ allows us to calculate the coordinates of a vector with respect to frame R_i in terms of its coordinates in frame R_j .

• **Example 2.1.** Deduce the matrices ${}^i\mathbf{T}_j$ and ${}^j\mathbf{T}_i$ from Figure 2.4. Using equation [2.5a], we directly obtain:

$${}^i\mathbf{T}_j = \begin{bmatrix} 0 & 0 & 1 & 3 \\ 0 & 1 & 0 & 12 \\ -1 & 0 & 0 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^j\mathbf{T}_i = \begin{bmatrix} 0 & 0 & -1 & 6 \\ 0 & 1 & 0 & -12 \\ 1 & 0 & 0 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

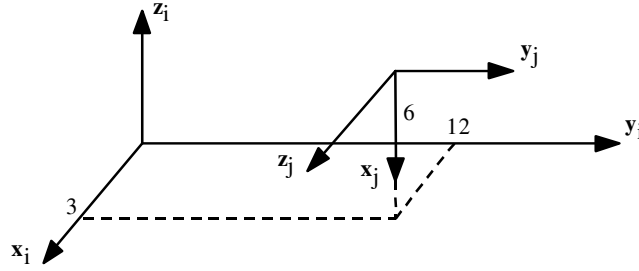


Figure 2.4. Example 2.1

2.3.3. Transformation of planes

The relative position of a point with respect to a plane is invariant with respect to the transformation applied to the set of {point, plane}. Thus:

$${}^j\mathbf{Q} {}^j\mathbf{P} = {}^i\mathbf{Q} {}^i\mathbf{P} = {}^i\mathbf{Q} {}^i\mathbf{T}_j {}^j\mathbf{P}$$

leading to:

$${}^j\mathbf{Q} = {}^i\mathbf{Q} {}^i\mathbf{T}_j \quad [2.7]$$

2.3.4. Transformation matrix of a pure translation

Let $\mathbf{Trans}(a, b, c)$ be this transformation, where a , b and c denote the translation along the x , y and z axes respectively. Since the orientation is invariant, the transformation $\mathbf{Trans}(a, b, c)$ is expressed as (Figure 2.5):

$${}^i\mathbf{T}_j = \mathbf{Trans}(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.8]$$

From now on, we will also use the notation $\mathbf{Trans}(\mathbf{u}, d)$ to denote a translation along an axis \mathbf{u} by a value d . Thus, the matrix $\mathbf{Trans}(a, b, c)$ can be decomposed into the product of three matrices $\mathbf{Trans}(\mathbf{x}, a)$ $\mathbf{Trans}(\mathbf{y}, b)$ $\mathbf{Trans}(\mathbf{z}, c)$, taking any order of the multiplication.

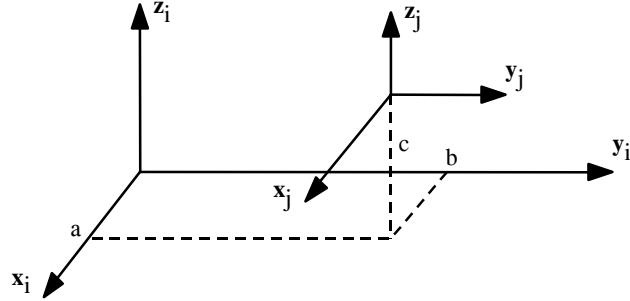


Figure 2.5. Transformation of pure translation

2.3.5. Transformation matrices of a rotation about the principle axes

2.3.5.1. Transformation matrix of a rotation about the \mathbf{x} axis by an angle θ

Let $\mathbf{Rot}(\mathbf{x}, \theta)$ be this transformation. From Figure 2.6, we deduce that the components of the unit vectors ${}^i\mathbf{s}_j$, ${}^i\mathbf{n}_j$, ${}^i\mathbf{a}_j$ along the axes \mathbf{x}_j , \mathbf{y}_j and \mathbf{z}_j respectively of frame R_j expressed in frame R_i are as follows:

$$\begin{cases} {}^i\mathbf{s}_j = [1 & 0 & 0 & 0]^T \\ {}^i\mathbf{n}_j = [0 & C\theta & S\theta & 0]^T \\ {}^i\mathbf{a}_j = [0 & -S\theta & C\theta & 0]^T \end{cases} \quad [2.9]$$

where $S\theta$ and $C\theta$ represent $\sin(\theta)$ and $\cos(\theta)$ respectively, and the superscript T indicates the transpose of the vector.

$${}^i\mathbf{T}_j = \mathbf{Rot}(\mathbf{x}, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta & -S\theta & 0 \\ 0 & S\theta & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & & & 0 \\ \mathbf{rot}(\mathbf{x}, \theta) & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.10]$$

where $\mathbf{rot}(\mathbf{x}, \theta)$ denotes the (3x3) orientation matrix.

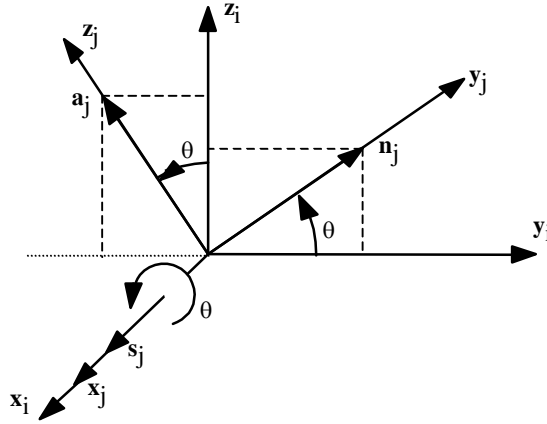


Figure 2.6. Transformation of a pure rotation about the x -axis

2.3.5.2. Transformation matrix of a rotation about the y axis by an angle θ

In the same way, we obtain:

$${}^i\mathbf{T}_j = \mathbf{Rot}(\mathbf{y}, \theta) = \begin{bmatrix} C\theta & 0 & S\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta & 0 & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & & & 0 \\ \mathbf{rot}(\mathbf{y}, \theta) & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.11]$$

2.3.5.3. Transformation matrix of a rotation θ about the z axis by an angle θ

We can also verify that:

$${}^i\mathbf{T}_j = \mathbf{Rot}(\mathbf{z}, \theta) = \begin{bmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & & & 0 \\ & \mathbf{rot}(\mathbf{z}, \theta) & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.12]$$

2.3.6. Properties of homogeneous transformation matrices

a) From equations [2.5], a transformation matrix can be written as:

$$\mathbf{T} = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & \mathbf{R} & \mathbf{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.13]$$

The matrix \mathbf{R} represents the rotation whereas the column matrix \mathbf{P} represents the translation. For a transformation of pure translation, $\mathbf{R} = \mathbf{I}_3$ (\mathbf{I}_3 represents the identity matrix of order 3), whereas $\mathbf{P} = \mathbf{0}$ for a transformation of pure rotation. The matrix \mathbf{R} represents the direction cosine matrix. It contains three independent parameters (one of the vectors \mathbf{s} , \mathbf{n} or \mathbf{a} can be deduced from the vector product of the other two, for example $\mathbf{s} = \mathbf{n} \times \mathbf{a}$; moreover, the dot product $\mathbf{n} \cdot \mathbf{a}$ is zero and the magnitudes of \mathbf{n} and \mathbf{a} are equal to 1).

b) The matrix \mathbf{R} is orthogonal, i.e. its inverse is equal to its transpose:

$$\mathbf{R}^{-1} = \mathbf{R}^T \quad [2.14]$$

c) The inverse of a matrix ${}^i\mathbf{T}_j$ defines the matrix ${}^j\mathbf{T}_i$.

To express the components of a vector ${}^i\mathbf{P}_1$ into frame R_j , we write:

$${}^j\mathbf{P}_1 = {}^j\mathbf{T}_i {}^i\mathbf{P}_1 \quad [2.15]$$

If we postmultiply equation [2.6] by ${}^i\mathbf{T}_j^{-1}$ (inverse of ${}^i\mathbf{T}_j$), we obtain:

$${}^i\mathbf{T}_j^{-1} {}^j\mathbf{P}_1 = {}^i\mathbf{P}_1 \quad [2.16]$$

From equations [2.15] and [2.16], we deduce that:

$${}^i\mathbf{T}_j^{-1} = {}^j\mathbf{T}_i \quad [2.17]$$

d) We can easily verify that:

$$\mathbf{Rot}^{-1}(\mathbf{u}, \theta) = \mathbf{Rot}(\mathbf{u}, -\theta) = \mathbf{Rot}(-\mathbf{u}, \theta) \quad [2.18]$$

$$\mathbf{Trans}^{-1}(\mathbf{u}, d) = \mathbf{Trans}(-\mathbf{u}, d) = \mathbf{Trans}(\mathbf{u}, -d) \quad [2.19]$$

e) The inverse of a transformation matrix represented by equation [2.13] can be obtained as:

$$\mathbf{T}^{-1} = \begin{bmatrix} & & -\mathbf{s}^T \mathbf{P} \\ & \mathbf{R}^T & -\mathbf{n}^T \mathbf{P} \\ & & -\mathbf{a}^T \mathbf{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & \mathbf{R}^T & -\mathbf{R}^T \mathbf{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.20]$$

f) Composition of two matrices. The multiplication of two transformation matrices gives a transformation matrix:

$$\mathbf{T}_1 \mathbf{T}_2 = \begin{bmatrix} \mathbf{R}_1 & \mathbf{P}_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_2 & \mathbf{P}_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \mathbf{R}_2 & \mathbf{R}_1 \mathbf{P}_2 + \mathbf{P}_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.21]$$

Note that the matrix multiplication is non-commutative ($\mathbf{T}_1 \mathbf{T}_2 \neq \mathbf{T}_2 \mathbf{T}_1$).

g) If a frame \mathbf{R}_0 is subjected to k consecutive transformations (Figure 2.7) and if each transformation i , ($i = 1, \dots, k$), is defined with respect to the current frame \mathbf{R}_{i-1} , then the transformation ${}^0\mathbf{T}_k$ can be deduced by multiplying all the transformation on the right as:

$${}^0\mathbf{T}_k = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 \dots {}^{k-1}\mathbf{T}_k \quad [2.22]$$

h) If a frame \mathbf{R}_j , defined by ${}^i\mathbf{T}_j$, undergoes a transformation \mathbf{T} that is defined relative to frame \mathbf{R}_i , then \mathbf{R}_j will be transformed into $\mathbf{R}_{j'}$ with ${}^i\mathbf{T}_{j'} = \mathbf{T} {}^i\mathbf{T}_j$ (Figure 2.8).

From the properties g and h, we deduce that:

- multiplication on the right (postmultiplication) of the transformation ${}^i\mathbf{T}_j$ indicates that the transformation is defined with respect to the current frame \mathbf{R}_j ;
- multiplication on the left (premultiplication) indicates that the transformation is defined with respect to the reference frame \mathbf{R}_i .

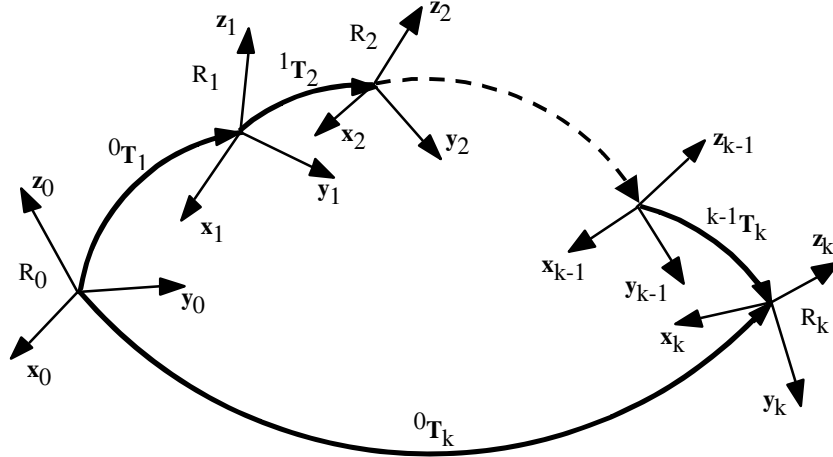


Figure 2.7. Composition of transformations: multiplication on the right

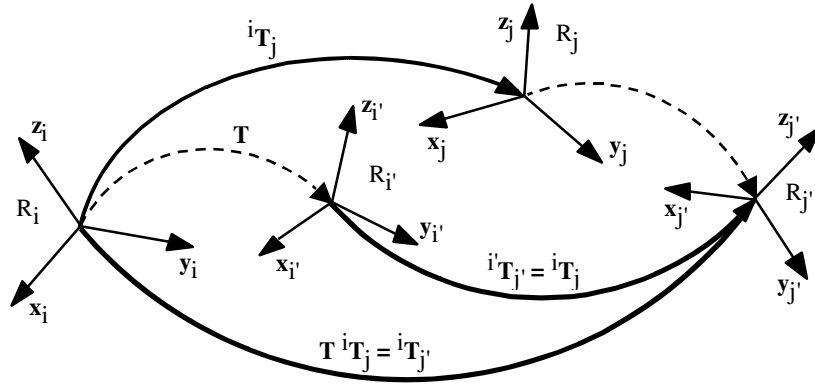


Figure 2.8. Composition of transformations: multiplication on the left

• **Example 2.2.** Consider the composite transformation illustrated in Figure 2.9 and defined by:

$${}^0T_2 = \text{Rot}(\mathbf{x}, \frac{\pi}{6}) \text{Trans}(\mathbf{y}, d)$$

- reading 0T_2 from left to right (Figure 2.9a): first, we apply the rotation; the new location of frame R_0 is denoted by frame R_1 ; then, the translation is defined with respect to frame R_1 ;

- reading 0T_2 from right to left (Figure 2.9b): first we apply the translation, then the rotation is defined with respect to frame R_0 .

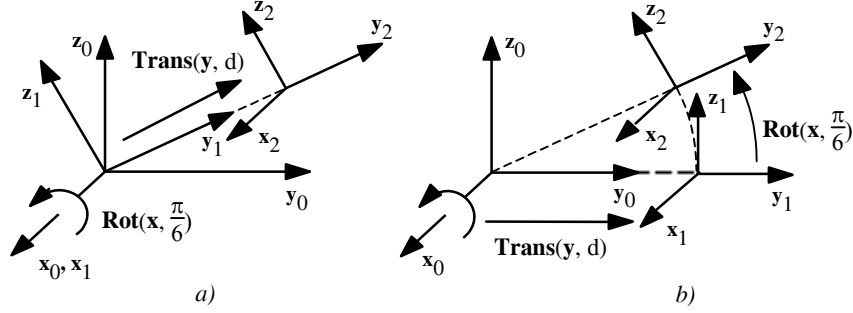


Figure 2.9. Example 2.2

- i) Consecutive transformations about the same axis. We note the following properties:

$$\mathbf{Rot}(\mathbf{u}, \theta_1) \mathbf{Rot}(\mathbf{u}, \theta_2) = \mathbf{Rot}(\mathbf{u}, (\theta_1 + \theta_2)) \quad [2.23]$$

$$\mathbf{Rot}(\mathbf{u}, \theta) \mathbf{Trans}(\mathbf{u}, d) = \mathbf{Trans}(\mathbf{u}, d) \mathbf{Rot}(\mathbf{u}, \theta) \quad [2.24]$$

- j) Decomposition of a transformation matrix. A transformation matrix can be decomposed into two transformation matrices, one represents a pure translation and the second a pure rotation:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.25]$$

2.3.7. Transformation matrix of a rotation about a general vector located at the origin

Let $\mathbf{Rot}(\mathbf{u}, \theta)$ be the transformation representing a rotation of an angle θ about an axis, with unit vector $\mathbf{u} = [u_x \ u_y \ u_z]^T$, located at the origin of frame R_i (Figure 2.10). We define the frame R_k such that \mathbf{z}_k is along the vector \mathbf{u} and \mathbf{x}_k is along the common normal between \mathbf{z}_k and \mathbf{z}_i . The matrix iT_k can be obtained as:

$${}^iT_k = \mathbf{Rot}(\mathbf{z}, \alpha) \mathbf{Rot}(\mathbf{x}, \beta) \quad [2.26]$$

where α is the angle between \mathbf{x}_i and \mathbf{x}_k about \mathbf{z}_i , and β is the angle between \mathbf{z}_i and \mathbf{u} about \mathbf{x}_k .

From equation [2.26], we obtain:

$$\mathbf{u} = {}^i\mathbf{a}_k = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} S\alpha S\beta \\ -C\alpha S\beta \\ C\beta \end{bmatrix} \quad [2.27]$$

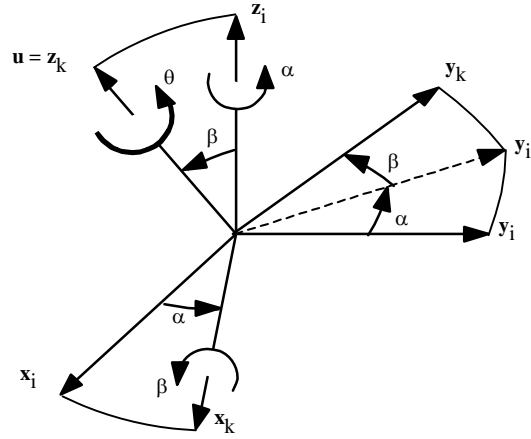


Figure 2.10. Transformation of pure rotation about any axis

The rotation about \mathbf{u} is equivalent to the rotation about \mathbf{z}_k . From properties g and h of § 2.3.6, we deduce that:

$$\mathbf{Rot}(\mathbf{u}, \theta) {}^i\mathbf{T}_k = {}^i\mathbf{T}_k \mathbf{Rot}(\mathbf{z}, \theta) \quad [2.28]$$

thus:

$$\begin{aligned} \mathbf{Rot}(\mathbf{u}, \theta) &= {}^i\mathbf{T}_k \mathbf{Rot}(\mathbf{z}, \theta) {}^i\mathbf{T}_k^{-1} \\ &= \mathbf{Rot}(\mathbf{z}, \alpha) \mathbf{Rot}(\mathbf{x}, \beta) \mathbf{Rot}(\mathbf{z}, \theta) \mathbf{Rot}(\mathbf{x}, -\beta) \mathbf{Rot}(\mathbf{z}, -\alpha) \end{aligned} \quad [2.29]$$

From this relation and using equation [2.27], we obtain:

$$\mathbf{Rot}(\mathbf{u}, \theta) = \begin{bmatrix} \mathbf{rot}(\mathbf{u}, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} u_x^2(1-C\theta)+C\theta & u_x u_y(1-C\theta)-u_z S\theta & u_x u_z(1-C\theta)+u_y S\theta & 0 \\ u_x u_y(1-C\theta)+u_z S\theta & u_y^2(1-C\theta)+C\theta & u_y u_z(1-C\theta)-u_x S\theta & 0 \\ u_x u_z(1-C\theta)-u_y S\theta & u_y u_z(1-C\theta)+u_x S\theta & u_z^2(1-C\theta)+C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.30]$$

We can easily remember this relation by writing it as:

$$\mathbf{rot}(\mathbf{u}, \theta) = \mathbf{u} \mathbf{u}^T (1 - C\theta) + \mathbf{I}_3 C\theta + \hat{\mathbf{u}} S\theta \quad [2.31]$$

where $\hat{\mathbf{u}}$ indicates the skew-symmetric matrix defined by the components of the vector \mathbf{u} such that:

$$\hat{\mathbf{u}} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \quad [2.32]$$

Note that the vector product $\mathbf{u} \times \mathbf{V}$ is obtained by $\hat{\mathbf{u}} \mathbf{V}$.

NOTES:-

- $\mathbf{Rot}(\mathbf{u}, \theta) = {}^i\mathbf{T}_k \mathbf{Rot}(\mathbf{z}, \theta) {}^i\mathbf{T}_k^{-1}$, can be explained by the fact that ${}^i\mathbf{T}_k \mathbf{Rot}(\mathbf{z}, \theta)$ gives frame R_k with respect to frame R_i in its initial configuration. To find frame R_i after $\mathbf{Rot}(\mathbf{u}, \theta)$, we have to multiply ${}^i\mathbf{T}_k \mathbf{Rot}(\mathbf{z}, \theta)$ by ${}^k\mathbf{T}_i$.

2.3.8. Equivalent angle and axis of a general rotation ¹

Let \mathbf{T} be any arbitrary rotational transformation matrix such that:

¹The Matlab function “ $\mathbf{U} = \text{vrrotmat2vec}(\mathbf{R})$ ” returns a 4x1 vector representing an axis-angle representation of rotation defined by the 3x3 matrix \mathbf{R} . The first three components of \mathbf{U} give the coordinates of the rotation axis and the fourth component gives the angle.

$$\mathbf{T} = \begin{bmatrix} s_x & n_x & a_x & 0 \\ s_y & n_y & a_y & 0 \\ s_z & n_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.33]$$

We solve the following expression for \mathbf{u} and θ :

$$\mathbf{Rot}(\mathbf{u}, \theta) = \mathbf{T} \quad \text{with } 0 \leq \theta \leq \pi$$

Adding the diagonal terms of equations [2.30] and [2.33], we obtain:

$$C\theta = \frac{1}{2}(s_x + n_y + a_z - 1) \quad [2.34]$$

From the off-diagonal terms, we obtain:

$$\begin{cases} 2u_x S\theta = n_z - a_y \\ 2u_y S\theta = a_x - s_z \\ 2u_z S\theta = s_y - n_x \end{cases} \quad [2.35]$$

yielding:

$$S\theta = \frac{1}{2} \sqrt{(n_z - a_y)^2 + (a_x - s_z)^2 + (s_y - n_x)^2} \quad [2.36]$$

From equations [2.34] and [2.36], we deduce that:

$$\theta = \text{atan2}(S\theta, C\theta) \quad \text{with } 0 \leq \theta \leq \pi \quad [2.37]$$

u_x , u_y and u_z are calculated using equation [2.35] if $S\theta \neq 0$. When $S\theta$ is small, the elements u_x , u_y and u_z cannot be determined with good accuracy by this equation. However, in the case where $C\theta < 0$, we obtain u_x , u_y and u_z more accurately using the diagonal terms of $\mathbf{Rot}(\mathbf{u}, \theta)$ as follows:

$$u_x = \pm \sqrt{\frac{s_x - C\theta}{1 - C\theta}}, u_y = \pm \sqrt{\frac{n_y - C\theta}{1 - C\theta}}, u_z = \pm \sqrt{\frac{a_z - C\theta}{1 - C\theta}} \quad [2.38]$$

From equation [2.35], we deduce that:

$$\begin{cases} u_x = \text{sign}(n_z - a_y) \sqrt{\frac{s_x - C\theta}{1 - C\theta}} \\ u_y = \text{sign}(a_x - s_z) \sqrt{\frac{n_y - C\theta}{1 - C\theta}} \\ u_z = \text{sign}(s_y - n_x) \sqrt{\frac{a_z - C\theta}{1 - C\theta}} \end{cases} \quad [2.39]$$

where $\text{sign}(\cdot)$ indicates the sign function of the expression between brackets, thus $\text{sign}(e) = +1$ if $e \geq 0$, and $\text{sign}(e) = -1$ if $e < 0$.

• **Example 2.3.** Suppose that the location of a frame R_E , which is fixed to the end-effector of a robot, relative to the reference frame R_0 is given by the matrix $\text{Rot}(\mathbf{x}, -\pi/4)$. Determine the vector ${}^E\mathbf{u}$ and the angle of rotation θ that transforms frame R_E to the location $\text{Rot}(\mathbf{y}, \pi/4) \text{Rot}(\mathbf{z}, \pi/2)$. We can write:

$$\text{Rot}(\mathbf{x}, -\pi/4) \text{Rot}(\mathbf{u}, \theta) = \text{Rot}(\mathbf{y}, \pi/4) \text{Rot}(\mathbf{z}, \pi/2)$$

Thus:

$$\text{Rot}(\mathbf{u}, \theta) = \text{Rot}(\mathbf{x}, \pi/4) \text{Rot}(\mathbf{y}, \pi/4) \text{Rot}(\mathbf{z}, \pi/2)$$

$$= \begin{bmatrix} 0 & -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/2 & -1/2 & 0 \\ 1/\sqrt{2} & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using equations [2.34] and [2.36], we get: $C\theta = -\frac{1}{2}$, $S\theta = \frac{\sqrt{3}}{2}$, giving $\theta = 2\pi/3$. Equation [2.35] yields: $u_x = \frac{1}{\sqrt{3}}$, $u_y = 0$, $u_z = \sqrt{\frac{2}{3}}$.

2.4. Representation of velocities

In this section, we will use the concept of screw to describe the velocity of a body in space.

2.4.1. Definition of a screw

A vector field, H , on \mathcal{R}^3 is a screw if there exist a point O_i and a vector $\mathbf{\Omega}$ such that for all points O_j in \mathcal{R}^3 :

$$\mathbf{H}_j = \mathbf{H}_i + \mathbf{\Omega} \times \mathbf{O}_i \mathbf{O}_j$$

where \mathbf{H}_j is the vector of H at O_j and the symbol \times indicates the vector product; $\mathbf{\Omega}$ is called the vector of the *screw* of H . The vector \mathbf{H}_j is also called the moment at O_j , whereas $\mathbf{\Omega}$ is also called the resultant of the screw.

Then, it is easy to prove that for every couple of points O_k and O_m :

$$\mathbf{H}_m = \mathbf{H}_k + \mathbf{\Omega} \times \mathbf{O}_k \mathbf{O}_m$$

Thus, the screw at a point O_i is well defined by the vectors \mathbf{H}_i and $\mathbf{\Omega}$, which can be concatenated in a single (6x1) vector.

2.4.2. Kinematic screw

Since the set of velocity vectors at all the points of a body defines a screw field, the screw at a point O_i can be defined by:

- \mathbf{v}_i representing the linear velocity at O_i with respect to the fixed frame R_0 , such that $\mathbf{v}_i = \frac{d}{dt}(\mathbf{O}_0 \mathbf{O}_i)$;
- $\mathbf{\omega}_i$ representing the angular velocity of the body with respect to frame R_0 . It constitutes the vector of the screw of the velocity vector field.

Thus, the velocity of a point O_j is calculated in terms of the velocity of the point O_i by the following equation:

$$\mathbf{v}_j = \mathbf{v}_i + \mathbf{\omega}_i \times \mathbf{O}_i \mathbf{O}_j \quad [2.40]$$

The components of \mathbf{v}_i and $\mathbf{\omega}_i$ can be concatenated to form the *kinematic screw* vector \mathbf{V}_i , i.e.:

$$\mathbf{V}_i = \begin{bmatrix} \mathbf{v}_i^T & \mathbf{\omega}_i^T \end{bmatrix}^T \quad [2.41]$$

The kinematic screw is also called *twist*.

2.4.3. Transformation matrices between screws

Let ${}^i\mathbf{v}_i$ and ${}^i\boldsymbol{\omega}_i$ be the vectors representing the kinematic screw in O_i , origin of frame R_i , expressed in frame R_i . To calculate ${}^j\mathbf{v}_j$ and ${}^j\boldsymbol{\omega}_j$ representing the kinematic screw in O_j expressed in frame R_j , we first note that:

$$\boldsymbol{\omega}_j = \boldsymbol{\omega}_i \quad [2.42]$$

$$\mathbf{v}_j = \mathbf{v}_i + \boldsymbol{\omega}_i \times \mathbf{L}_{i,j} \quad [2.43]$$

$\mathbf{L}_{i,j}$ being the position vector connecting O_i to O_j .

Equations [2.42] and [2.43] can be rewritten as:

$$\begin{bmatrix} \mathbf{v}_j \\ \boldsymbol{\omega}_j \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & -\hat{\mathbf{L}}_{i,j} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \quad [2.44]$$

where \mathbf{I}_3 and $\mathbf{0}_3$ represent the (3x3) identity matrix and zero matrix respectively. Projecting this relation in frame R_i , we obtain:

$$\begin{bmatrix} {}^i\mathbf{v}_j \\ {}^i\boldsymbol{\omega}_j \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & -{}^i\hat{\mathbf{P}}_j \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} {}^i\mathbf{v}_i \\ {}^i\boldsymbol{\omega}_i \end{bmatrix} \quad [2.45]$$

Since ${}^j\mathbf{v}_j = {}^j\mathbf{R}_i {}^i\mathbf{v}_j$ and ${}^j\boldsymbol{\omega}_j = {}^j\mathbf{R}_i {}^i\boldsymbol{\omega}_j$, equation [2.45] gives:

$${}^j\mathbf{V}_j = {}^j\mathbf{S}_i {}^i\mathbf{V}_i \quad [2.46]$$

where ${}^j\mathbb{T}_i$ is the (6x6) transformation matrix between screws:

$${}^j\mathbf{S}_i = \begin{bmatrix} {}^j\mathbf{R}_i & -{}^j\mathbf{R}_i {}^i\hat{\mathbf{P}}_j \\ \mathbf{0}_3 & {}^j\mathbf{R}_i \end{bmatrix} \quad [2.47]$$

The transformation matrices between screws have the following properties:

i) product:

$${}^0\mathbf{S}_j = {}^0\mathbf{S}_1 {}^1\mathbf{S}_2 \dots {}^{j-1}\mathbf{S}_j \quad [2.48]$$

ii) *inverse*:

$${}^i\mathbf{S}_j^{-1} = \begin{bmatrix} {}^i\mathbf{R}_j & {}^i\hat{\mathbf{P}}_j {}^i\mathbf{R}_j \\ \mathbf{0}_3 & {}^i\mathbf{R}_j \end{bmatrix} = {}^j\mathbf{S}_i \quad [2.49]$$

Note that equation [2.49] gives another possibility, other than equation [2.45], to define the transformation matrix between screws.

From [2.49] and [2.49], we can write:

$${}^i\mathbf{S}_j = \begin{bmatrix} {}^i\mathbf{R}_j & {}^i\hat{\mathbf{P}}_j {}^i\mathbf{R}_j \\ \mathbf{0}_3 & {}^i\mathbf{R}_j \end{bmatrix} = \begin{bmatrix} {}^i\mathbf{R}_j & -{}^i\mathbf{R}_j {}^j\hat{\mathbf{P}}_i \\ \mathbf{0}_3 & {}^i\mathbf{R}_j \end{bmatrix}$$

2.5. Differential translation and rotation of frames

The differential transformation of the position and orientation – or location – of a frame R_i attached to any body may be expressed by a differential translation vector $d\mathbf{P}_i$ expressing the translation of the origin of frame R_i , and of a differential rotation vector δ_i , equal to $\mathbf{u}_i d\theta$, representing the rotation of an angle $d\theta$ about an axis, with unit vector \mathbf{u}_i , passing through the origin O_i .

Given a transformation ${}^i\mathbf{T}_j$, the transformation ${}^i\mathbf{T}_j + d{}^i\mathbf{T}_j$ can be calculated, taking into account the property h of § 2.3.6, by the premultiplication rule as:

$${}^i\mathbf{T}_j + d{}^i\mathbf{T}_j = \mathbf{Trans}({}^i dx_i, {}^i dy_i, {}^i dz_i) \mathbf{Rot}({}^i \mathbf{u}_i, d\theta) {}^i\mathbf{T}_j \quad [2.50]$$

Thus, the differential of ${}^i\mathbf{T}_j$ is equal to:

$$d{}^i\mathbf{T}_j = [\mathbf{Trans}({}^i dx_i, {}^i dy_i, {}^i dz_i) \mathbf{Rot}({}^i \mathbf{u}_i, d\theta) - \mathbf{I}_4] {}^i\mathbf{T}_j \quad [2.51]$$

In the same way, the transformation ${}^i\mathbf{T}_j + d{}^i\mathbf{T}_j$ can be calculated, using the postmultiplication rule as:

$${}^i\mathbf{T}_j + d{}^i\mathbf{T}_j = {}^i\mathbf{T}_j \mathbf{Trans}({}^j dx_j, {}^j dy_j, {}^j dz_j) \mathbf{Rot}({}^j \mathbf{u}_j, d\theta) \quad [2.52]$$

and the differential of ${}^i\mathbf{T}_j$ becomes:

$$\mathbf{d}^i \mathbf{T}_j = {}^i \mathbf{T}_j [\mathbf{Trans}(jdx_j, jdy_j, jdz_j) \mathbf{Rot}(j\mathbf{u}_j, d\theta) - \mathbf{I}_4] \quad [2.53]$$

From equations [2.51] and [2.53], the differential transformation matrix Δ is defined as [Paul 81]:

$$\Delta = [\mathbf{Trans}(dx, dy, dz) \mathbf{Rot}(\mathbf{u}, d\theta) - \mathbf{I}_4] \quad [2.54]$$

such that:

$$\mathbf{d}^i \mathbf{T}_j = {}^i \Delta {}^i \mathbf{T}_j \quad [2.55]$$

or:

$$\mathbf{d}^i \mathbf{T}_j = {}^i \mathbf{T}_j j\Delta \quad [2.56]$$

Assuming that $d\theta$ is sufficiently small so that $S(d\theta) \approx d\theta$ and $C(d\theta) \approx 1$, the transformation matrix of a pure rotation $d\theta$ about an axis of unit vector \mathbf{u} can be calculated from equations [2.30] and [2.54] as:

$$j\Delta = \begin{bmatrix} j\hat{\delta}_j & j\mathbf{dP}_j \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} j\hat{\mathbf{u}}_j d\theta & j\mathbf{dP}_j \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad [2.57]$$

where $\hat{\mathbf{u}}$ and $\hat{\delta}$ represent the skew-symmetric matrices defined by the vectors \mathbf{u} and δ respectively.

Note that the transformation matrix between screws can also be used to transform the differential translation and rotation vectors between frames:

$$\begin{bmatrix} j\mathbf{dP}_j \\ j\hat{\delta}_j \end{bmatrix} = j\mathbf{S}_i \begin{bmatrix} {}^i \mathbf{dP}_i \\ {}^i \hat{\delta}_i \end{bmatrix} \quad [2.58]$$

In a similar way as for the kinematic screw, we call the concatenation of \mathbf{dP}_i and $\hat{\delta}_i$ the *differential screw*.

• **Example 2.4.** Consider using the differential model of a robot to control its displacement. The differential model calculates the joint increments corresponding to the desired elementary displacement of frame R_n fixed to the terminal link (Figure 2.11). However, the task of the robot is often described in the tool frame R_E ,

which is also fixed to the terminal link. The problem is to calculate ${}^n\mathbf{dP}_n$ and ${}^n\delta_n$ in terms of ${}^E\mathbf{dP}_E$ and ${}^E\delta_E$.

Let the transformation describing the tool frame in frame R_n be:

$${}^n\mathbf{T}_E = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.1 \\ 0 & 0 & 1 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and that the value of the desired elementary displacement is:

$${}^E\mathbf{dP}_E = [0 \quad 0 \quad -0.01]^T, \quad {}^E\delta_E = [0 \quad -0.05 \quad 0]^T$$

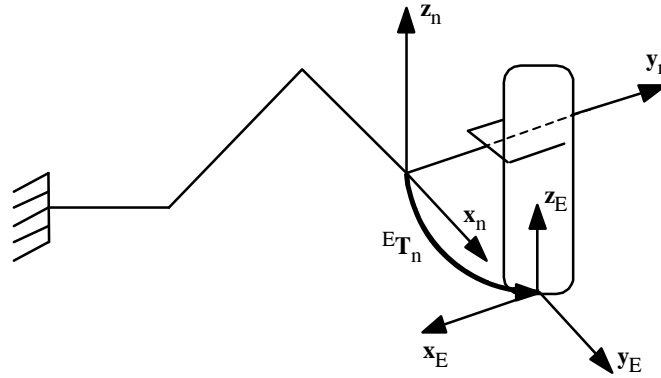


Figure 2.11. Example 2.4

Using equation [2.58], we obtain:

$${}^n\delta_n = {}^n\mathbf{R}_E {}^E\delta_E, \quad {}^n\mathbf{dP}_n = {}^n\mathbf{R}_E ({}^E\delta_E \times {}^E\mathbf{P}_n + {}^E\mathbf{dP}_E)$$

The numerical application gives:

$${}^n\mathbf{dP}_n = [0 \quad 0.015 \quad -0.005]^T, \quad {}^n\delta_n = [-0.05 \quad 0 \quad 0]^T$$

In a similar way, we can evaluate the error in the location of the tool frame due to errors in the position and orientation of the terminal frame. Suppose that the position error is equal to 10 mm in all directions and that the rotation error is estimated as 0.01 radian about the x axis:

$${}^n\mathbf{dP}_n = [0.01 \quad 0.01 \quad 0.01]^T, \quad {}^n\boldsymbol{\delta}_n = [0.01 \quad 0 \quad 0]^T$$

The error on the tool frame is calculated by:

$${}^E\boldsymbol{\delta}_E = {}^E\mathbf{R}_n {}^n\boldsymbol{\delta}_n, \quad {}^E\mathbf{dP}_E = {}^E\mathbf{R}_n ({}^n\boldsymbol{\delta}_n \times {}^n\mathbf{P}_E + {}^n\mathbf{dP}_n)$$

which results in:

$${}^E\mathbf{dP}_E = [-0.013 \quad 0.01 \quad 0.011]^T, \quad {}^E\boldsymbol{\delta}_E = [0 \quad 0.01 \quad 0]^T$$

2.6. Representation of forces (wrench)

A collection of forces and moments acting on a body can be reduced to a *wrench* \mathbf{F}_i at point O_i , which is composed of a force \mathbf{f}_i at O_i and a moment \mathbf{m}_i about O_i :

$$\mathbf{F}_i = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{m}_i \end{bmatrix} \quad [2.59]$$

Note that the vector field of the moments constitutes a screw where the vector of the screw is \mathbf{f}_i . Thus, the wrench forms a screw.

Consider a given wrench ${}^i\mathbf{F}_i$, expressed in frame R_i . For calculating the equivalent wrench ${}^j\mathbf{F}_j$, we use the transformation matrix between screws such that:

$$\begin{bmatrix} {}^j\mathbf{m}_j \\ {}^j\mathbf{f}_j \end{bmatrix} = {}^j\mathbf{S}_i \begin{bmatrix} {}^i\mathbf{m}_i \\ {}^i\mathbf{f}_i \end{bmatrix} \quad [2.60]$$

which gives:

$${}^j\mathbf{f}_j = {}^j\mathbf{R}_i {}^i\mathbf{f}_i \quad [2.61]$$

$${}^j\mathbf{m}_j = {}^j\mathbf{R}_i ({}^i\mathbf{f}_i \times {}^i\mathbf{P}_j + {}^i\mathbf{m}_i) \quad [2.62]$$

It is often more practical to permute the order of \mathbf{f}_i and \mathbf{m}_i . In this case, equation [2.60] becomes:

$$\begin{bmatrix} {}^j\mathbf{f}_j \\ {}^j\mathbf{m}_j \end{bmatrix} = {}^i\mathbf{S}_j^T \begin{bmatrix} {}^i\mathbf{f}_i \\ {}^i\mathbf{m}_i \end{bmatrix} \quad [2.63]$$

• **Example 2.5.** Let the transformation matrix ${}^n\mathbf{T}_E$ describing the location of the tool frame with respect to the terminal frame be:

$${}^n\mathbf{T}_E = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.1 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Supposing that we want to exert a wrench ${}^E\mathbf{F}_E$ with this tool such that ${}^E\mathbf{f}_E = [0 \ 0 \ 5]^T$ and ${}^E\mathbf{m}_E = [0 \ 0 \ 3]^T$, determine the corresponding wrench ${}^n\mathbf{F}_n$ at the origin O_n and referred to frame R_n . Using equations [2.61] and [2.62], it follows that:

$$\begin{aligned} {}^n\mathbf{f}_n &= {}^n\mathbf{R}_E {}^E\mathbf{f}_E \\ {}^n\mathbf{m}_n &= {}^n\mathbf{R}_E ({}^E\mathbf{f}_E \times {}^E\mathbf{P}_n + {}^E\mathbf{m}_E) \end{aligned}$$

The numerical application leads to:

$$\begin{aligned} {}^n\mathbf{f}_n &= [0 \ 0 \ 5]^T \\ {}^n\mathbf{m}_n &= [0.5 \ 0 \ 3]^T \end{aligned}$$

2.7. Conclusion

In the first part of this chapter, we have developed the homogeneous transformation matrix. This notation constitutes the basic tool for the modeling of robots and their environment. Other techniques have been used in robotics: quaternion [Yang 66], [Castelain 86], (3x3) rotation matrices [Coiffet 81] and the Rodrigues formulation [Wang 83]. Readers interested in these techniques can consult the given references.

We have also recalled some definitions about screws, and transformation matrices between screws, as well as differential transformations. These concepts will be used extensively in this book. In the following chapter, we deal with the problem of robot modeling.

Chapter 3

Direct geometric model of serial robots

3.1. Introduction

The design and control of a robot requires the computation of some mathematical models such as:

- transformation models between the joint space (in which the configuration of the robot is defined) and the task space (in which the location of the end-effector is specified). These transformation models are very important since robots are controlled in the joint space, whereas tasks are defined in the task space. Two classes of models are considered:
 - direct and inverse geometric models, which give the location of the end-effector as a function of the joint variables of the mechanism and vice versa;
 - direct and inverse kinematic models, which give the velocity of the end-effector as a function of the joint velocities and vice versa;
- dynamic models giving the relations between the input torques or forces of the actuators and the positions, velocities and accelerations of the joints.

The automatic symbolic computation of these models has largely been addressed in the literature [Dillon 73], [Khalil 76], [Zabala 78], [Kreuzer 79], [Aldon 82], [Cesareo 84], [Megahed 84], [Murray 84], [Kircanski 85], [Burdick 86], [Izaguirre 86], [Khalil 89a]. The algorithms presented in this book have been used in the development of the software package SYMORO+ [Khalil 97], which deals with all the above-mentioned models.

The modeling of robots in a systematic and automatic way requires an adequate method for the description of their structure. Several methods and notations have been proposed [Denavit 55], [Sheth 71], [Renaud 75], [Khalil 76], [Borrel 79], [Craig 86a]. The most popular among these is the Denavit-Hartenberg method

[Denavit 55]. This method is developed for serial structures and presents ambiguities when applied to robots with closed or tree chains. For this reason, we will use the notation of Khalil and Kleinfinger [Khalil 86a], which gives a unified description for all mechanical articulated systems, including mobile robots [Venture 2006], with a minimum number of parameters. In this chapter, we will present the geometric description and the direct geometric model of serial robots. Tree and closed loop structures will be covered in Chapter 7.

3.2. Description of the geometry of serial robots

A serial robot is composed of a sequence of $n + 1$ links and n joints. The links are assumed to be perfectly rigid. The links are numbered such that link 0 constitutes the base of the robot and link n is the terminal link (Figure 3.1). Joint j connects link j to link $j - 1$ and its variable is denoted q_j . The joints are either revolute or prismatic and are assumed to be ideal (no backlash, no elasticity). A complex joint can be represented by an equivalent combination of revolute and prismatic joints with zero-length massless links. In order to define the relationship between the location of links, we assign a frame R_j attached to each link j , such that:

- the z_j axis is along the axis of joint j ;
- the x_j axis is aligned with the common normal between z_j and z_{j+1} :
 - . if z_j and z_{j+1} are collinear x_j is not unique, it can be taken in any plane perpendicular to them,
 - . if z_j and z_{j+1} are parallel, x_j is not unique, it is in the plane defined by them,
 - . in the case of intersecting joint axes, x_j is normal to the plane defined by them and passing through their intersection point;
- the intersection of x_j and z_j defines the origin O_j , the y_j axis is formed by the right-hand rule to complete the coordinate system (x_j, y_j, z_j) .

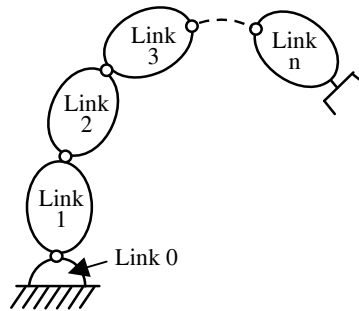
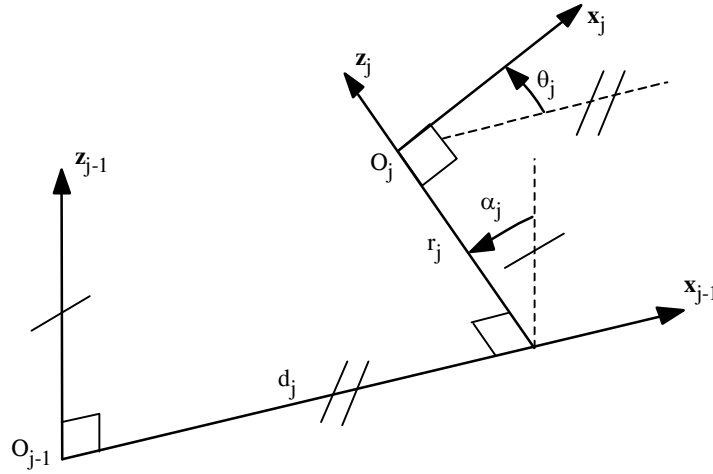


Figure 3.1. Robot with simple open structure

The transformation matrix from frame R_{j-1} to frame R_j is expressed as a function of the following four geometric parameters (Figure 3.2):

- α_j : the angle between \mathbf{z}_{j-1} and \mathbf{z}_j about \mathbf{x}_{j-1} ;
- d_j : the distance between \mathbf{z}_{j-1} and \mathbf{z}_j along \mathbf{x}_{j-1} ;
- θ_j : the angle between \mathbf{x}_{j-1} and \mathbf{x}_j about \mathbf{z}_j ;
- r_j : the distance between \mathbf{x}_{j-1} and \mathbf{x}_j along \mathbf{z}_j .

**Figure 3.2.** The geometric parameters in the case of a simple open structure

The variable of joint j , defining the relative orientation or position between links $j-1$ and j , is either θ_j or r_j , depending on whether the joint is revolute or prismatic respectively. This is defined by the relation:

$$q_j = \bar{\sigma}_j \theta_j + \sigma_j r_j \quad [3.1a]$$

with:

- $\sigma_j = 0$ if joint j is revolute;
- $\sigma_j = 1$ if joint j is prismatic;
- $\bar{\sigma}_j = 1 - \sigma_j$.

By analogy, we define the parameter \bar{q}_j by:

$$\bar{q}_j = \sigma_j \theta_j + \bar{\sigma}_j r_j \quad [3.1b]$$

The transformation matrix defining frame R_j relative to frame R_{j-1} is given as (Figure 3.2):

$$\begin{aligned} {}^{j-1}T_j &= \mathbf{Rot}(\mathbf{x}, \alpha_j) \mathbf{Trans}(\mathbf{x}, d_j) \mathbf{Rot}(\mathbf{z}, \theta_j) \mathbf{Trans}(\mathbf{z}, r_j) \\ &= \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad [3.2]$$

We note that the (3x3) rotation matrix ${}^{j-1}\mathbf{R}_j$ can be obtained as:

$${}^{j-1}\mathbf{R}_j = \mathbf{rot}(\mathbf{x}, \alpha_j) \mathbf{rot}(\mathbf{z}, \theta_j) \quad [3.3]$$

The transformation matrix defining frame R_{j-1} relative to frame R_j is given as:

$$\begin{aligned} {}^jT_{j-1} &= \mathbf{Trans}(\mathbf{z}, -r_j) \mathbf{Rot}(\mathbf{z}, -\theta_j) \mathbf{Trans}(\mathbf{x}, -d_j) \mathbf{Rot}(\mathbf{x}, -\alpha_j) \\ &= \begin{bmatrix} & & -d_j C\theta_j \\ & {}^{j-1}\mathbf{R}_j^T & d_j S\theta_j \\ & & -r_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad [3.4]$$

NOTES.—

- the frame R_0 is chosen to be aligned with frame R_1 when $q_1 = 0$. This means that \mathbf{z}_0 is aligned with \mathbf{z}_1 , whereas the origin O_0 is coincident with the origin O_1 if joint 1 is revolute, and \mathbf{x}_0 is parallel to \mathbf{x}_1 if joint 1 is prismatic. This choice makes $\alpha_1 = 0$, $d_1 = 0$ and $\bar{q}_1 = 0$;
- in a similar way, the choice of the \mathbf{x}_n axis to be aligned with \mathbf{x}_{n-1} when $q_n = 0$ makes $\bar{q}_n = 0$;
- if joint j is prismatic, the \mathbf{z}_j axis must be taken to be parallel to the joint axis but can have any position in space. So, we place it in such a way that $d_j = 0$ or $d_{j+1} = 0$;
- if \mathbf{z}_j is parallel to \mathbf{z}_{j+1} , we place \mathbf{x}_j in such a way that $r_j = 0$ or $r_{j+1} = 0$;

- assuming that each joint is driven by an independent actuator, the vector of joint variables \mathbf{q} can be obtained from the vector of encoder readings \mathbf{q}_c using the relation:

$$\mathbf{q} = \mathbf{K} \mathbf{q}_c + \mathbf{q}_0$$

where \mathbf{K} is an $(n \times n)$ constant matrix and \mathbf{q}_0 is an offset vector representing the robot configuration when $\mathbf{q}_c = \mathbf{0}$;

- if a chain contains two or more consecutive parallel joints, the transformation matrices between them can be reduced to one equivalent transformation matrix using the sum of the joint variables. For example, if $\alpha_{j+1} = 0$, i.e. if \mathbf{z}_j and \mathbf{z}_{j+1} are parallel, the transformation ${}^{j-1}\mathbf{T}_{j+1}$ is written as:

$$\begin{aligned} {}^{j-1}\mathbf{T}_{j+1} &= {}^{j-1}\mathbf{T}_j {}^j\mathbf{T}_{j+1} = \mathbf{Rot}(\mathbf{x}, \alpha_j) \mathbf{Trans}(\mathbf{x}, d_j) \mathbf{Rot}(\mathbf{z}, \theta_j) \mathbf{Trans}(\mathbf{z}, r_j) \\ &\quad \mathbf{Trans}(\mathbf{x}, d_{j+1}) \mathbf{Rot}(\mathbf{z}, \theta_{j+1}) \mathbf{Trans}(\mathbf{z}, r_{j+1}) \quad [3.5] \\ &= \\ &\quad \begin{bmatrix} C(\theta_j + \theta_{j+1}) & -S(\theta_j + \theta_{j+1}) & 0 & d_j + d_{j+1}C\theta_j \\ C\alpha_j S(\theta_j + \theta_{j+1}) & C\alpha_j C(\theta_j + \theta_{j+1}) & -S\alpha_j & d_{j+1}C\alpha_j S\theta_j - (r_j + r_{j+1})S\alpha_j \\ S\alpha_j S(\theta_j + \theta_{j+1}) & S\alpha_j C(\theta_j + \theta_{j+1}) & C\alpha_j & d_{j+1}S\alpha_j S\theta_j + (r_j + r_{j+1})C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

and the inverse transformation has the expression:

$${}^{j+1}\mathbf{T}_{j-1} = \begin{bmatrix} & & -d_j C(\theta_j + \theta_{j+1}) - d_{j+1} C\theta_{j+1} \\ & {}^{j-1}\mathbf{R}_{j+1}^T & d_j S(\theta_j + \theta_{j+1}) + d_{j+1} S\theta_{j+1} \\ & & -(r_j + r_{j+1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [3.6]$$

The above expressions contain terms in $(\theta_j + \theta_{j+1})$ and $(r_j + r_{j+1})$. This result can be generalized for the case of multiple consecutive parallel axes [Kleininger 86a].

• **Example 3.1.** Geometric description of the Stäubli RX-90 robot (Figure 3.3a). The shoulder is of RRR type and the wrist has three revolute joints whose axes intersect at a point (Figure 3.3b). From a methodological point of view, we first place the \mathbf{z}_j axes on the joint axes, then the \mathbf{x}_j axes according to the previously mentioned conventions. Then, we determine the geometric parameters defining each frame \mathbf{R}_j

with respect to frame R_{j-1} . The link coordinate frames are indicated in Figure 3.3b and the geometric parameters are given in Table 3.1.

Table 3.1. *Geometric parameters of the Stäubli RX-90 robot*

j	σ_j	α_j	d_j	θ_j	r_j
1	0	0	0	θ_1	0
2	0	$\pi/2$	0	θ_2	0
3	0	0	D3	θ_3	0
4	0	$-\pi/2$	0	θ_4	RL4
5	0	$\pi/2$	0	θ_5	0
6	0	$-\pi/2$	0	θ_6	0



Figure 3.3a. *General view of the Stäubli RX-90 robot*
(Courtesy of Stäubli company)

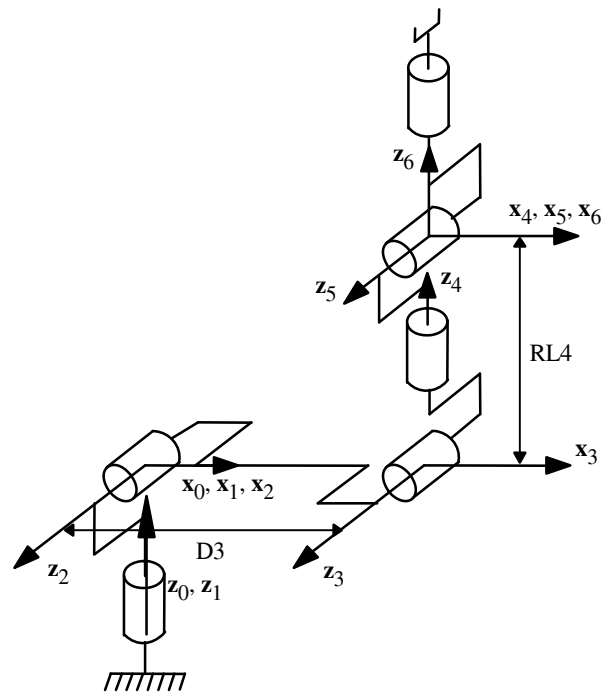


Figure 3.3b. Link coordinate frames for the Stäubli RX-90 robot

- **Example 3.2.** Geometric description of a SCARA robot (Figure 3.4). The geometric parameters of a four degree-of-freedom SCARA robot are given in Table 3.2.

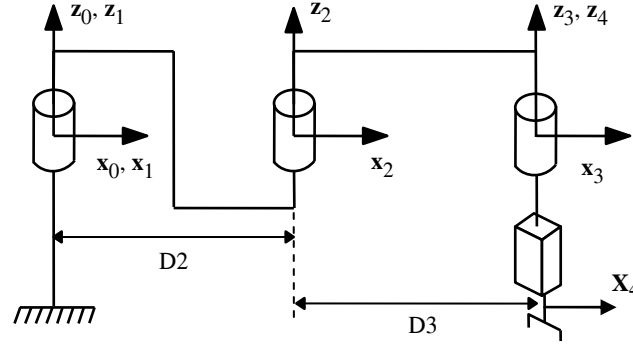


Figure 3.4. SCARA Robot

Table 3.2. Geometric parameters of a SCARA robot

j	σ_j	α_j	d_j	θ_j	r_j
1	0	0	0	θ_1	0
2	0	0	D2	θ_2	0
3	0	0	D3	θ_3	0
4	1	0	0	0	r_4

3.3. Direct geometric model

The Direct Geometric Model (DGM) is the set of relations that defines the location of the end-effector of the robot as a function of its joint coordinates. For a serial structure, it may be represented by the transformation matrix 0T_n as:

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n) \quad [3.7]$$

This relation can be numerically computed using the general transformation matrix ${}^{j-1}T_j$ given by equation [3.2], or symbolically derived after substituting the values of the constant geometric parameters in the transformation matrices (Example 3.3). The symbolic method needs less computational operations.

The direct geometric model of a robot may also be represented by the relation:

$$\mathbf{X} = \mathbf{f}(\mathbf{q}) \quad [3.8]$$

where \mathbf{q} is the vector of joint variables such that:

$$\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T \quad [3.9]$$

The position and orientation of the terminal link are defined as:

$$\mathbf{X} = [x_1 \ x_2 \ \dots \ x_m]^T \quad [3.10]$$

There are several possibilities of defining the vector \mathbf{X} as we will see in § 3.6. For example, with the elements of the matrix ${}^0\mathbf{T}_n$:

$$\mathbf{X} = [P_x \ P_y \ P_z \ s_x \ s_y \ s_z \ n_x \ n_y \ n_z \ a_x \ a_y \ a_z]^T \quad [3.11]$$

Taking into account that $\mathbf{s} = \mathbf{n} \times \mathbf{a}$, we can also take:

$$\mathbf{X} = [P_x \ P_y \ P_z \ n_x \ n_y \ n_z \ a_x \ a_y \ a_z]^T \quad [3.12]$$

• **Example 3.3.** Symbolic direct geometric model of the Stäubli RX-90 robot (Figure 3.3). From Table 3.1 and using equation [3.2], we write the elementary transformation matrices ${}^j\mathbf{T}_j$ as:

$${}^0\mathbf{T}_1 = \begin{bmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^1\mathbf{T}_2 = \begin{bmatrix} C2 & -S2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S2 & C2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^2\mathbf{T}_3 = \begin{bmatrix} C3 & -S3 & 0 & D3 \\ S3 & C3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since the joint axes 2 and 3 are parallel, we can write the transformation matrix ${}^1\mathbf{T}_3$ using equation [3.5] as:

$${}^1\mathbf{T}_3 = \begin{bmatrix} C23 & -S23 & 0 & C2D3 \\ 0 & 0 & -1 & 0 \\ S23 & C23 & 0 & S2D3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

with $C23 = \cos(\theta_2 + \theta_3)$ and $S23 = \sin(\theta_2 + \theta_3)$.

$${}^3\mathbf{T}_4 = \begin{bmatrix} C4 & -S4 & 0 & 0 \\ 0 & 0 & 1 & RL4 \\ -S4 & -C4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^4\mathbf{T}_5 = \begin{bmatrix} C5 & S5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S5 & C5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^5\mathbf{T}_6 = \begin{bmatrix} C6 & -S6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S6 & C6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In order to compute ${}^0\mathbf{T}_6$, it is better to multiply the matrices ${}^j\mathbf{T}_j$ starting from the last transformation matrix and working back to the base, mainly for two reasons:

- the intermediate matrices ${}^j\mathbf{T}_6$, denoted as \mathbf{U}_j , will be used to obtain the inverse geometric model (Chapter 4);
- this reduces the number of operations (additions and multiplications) of the model.

We thus compute successively \mathbf{U}_j for $j = 5, \dots, 0$:

$$\mathbf{U}_5 = {}^5\mathbf{T}_6$$

$$\mathbf{U}_4 = {}^4\mathbf{T}_6 = {}^4\mathbf{T}_5 \mathbf{U}_5 = \begin{bmatrix} C5C6 & -C5S6 & -S5 & 0 \\ S6 & C6 & 0 & 0 \\ S5C6 & -S5S6 & C5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{U}_3 = {}^3\mathbf{T}_6 = {}^3\mathbf{T}_4 \mathbf{U}_4 = \begin{bmatrix} C4C5C6 - S4S6 & -C4C5S6 - S4C6 & -C4S5 & 0 \\ S5C6 & -S5S6 & C5 & RL4 \\ -S4C5C6 - C4S6 & S4C5S6 - C4C6 & S4S5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{U}_2 = {}^2\mathbf{T}_6 = {}^2\mathbf{T}_3 \mathbf{U}_3$$

The \mathbf{s} , \mathbf{n} , \mathbf{a} , \mathbf{P} vectors of \mathbf{U}_2 are:

$$\begin{aligned} s_x &= C3(C4C5C6 - S4S6) - S3S5C6 \\ s_y &= S3(C4C5C6 - S4S6) + C3S5C6 \\ s_z &= -S4C5C6 - C4S6 \\ n_x &= -C3(C4C5S6 + S4C6) + S3S5S6 \\ n_y &= -S3(C4C5S6 + S4C6) - C3S5S6 \\ n_z &= S4C5S6 - C4C6 \\ a_x &= -C3C4S5 - S3C5 \\ a_y &= -S3C4S5 + C3C5 \\ a_z &= S4S5 \\ P_x &= -S3RL4 + D3 \end{aligned}$$

$$P_y = C3RL4$$

$$P_z = 0$$

$$U_1 = {}^1T_6 = {}^1T_2 U_2 = {}^1T_3 U_3$$

The corresponding **s**, **n**, **a**, **P** vectors are:

$$s_x = C23(C4C5C6 - S4S6) - S23S5C6$$

$$s_y = S4C5C6 + C4S6$$

$$s_z = S23(C4C5C6 - S4S6) + C23S5C6$$

$$n_x = -C23(C4C5S6 + S4C6) + S23S5S6$$

$$n_y = -S4C5S6 + C4C6$$

$$n_z = -S23(C4C5S6 + S4C6) - C23S5S6$$

$$a_x = -C23C4S5 - S23C5$$

$$a_y = -S4S5$$

$$a_z = -S23C4S5 + C23C5$$

$$P_x = -S23 RL4 + C2D3$$

$$P_y = 0$$

$$P_z = C23 RL4 + S2D3$$

Finally:

$$U_0 = {}^0T_6 = {}^0T_1 U_1$$

The corresponding **s**, **n**, **a**, **P** vectors are:

$$s_x = C1(C23(C4C5C6 - S4S6) - S23S5C6) - S1(S4C5C6 + C4S6)$$

$$s_y = S1(C23(C4C5C6 - S4S6) - S23S5C6) + C1(S4C5C6 + C4S6)$$

$$s_z = S23(C4C5C6 - S4S6) + C23S5C6$$

$$n_x = C1(-C23(C4C5S6 + S4C6) + S23S5S6) + S1(S4C5S6 - C4C6)$$

$$n_y = S1(-C23(C4C5S6 + S4C6) + S23S5S6) - C1(S4C5S6 - C4C6)$$

$$n_z = -S23(C4C5S6 + S4C6) - C23S5S6$$

$$a_x = -C1(C23C4S5 + S23C5) + S1S4S5$$

$$a_y = -S1(C23C4S5 + S23C5) - C1S4S5$$

$$a_z = -S23C4S5 + C23C5$$

$$P_x = -C1(S23 RL4 - C2D3)$$

$$P_y = -S1(S23 RL4 - C2D3)$$

$$P_z = C23 RL4 + S2D3$$

3.4. Optimization of the computation of the direct geometric model

The control of a robot manipulator requires fast computation of its different models. An efficient method to reduce the computation time is to generate a symbolic customized model for each specific robot. To obtain this model, we expand the matrix multiplications to transform them into scalar equations. Each element of a matrix containing at least one mathematical operation is replaced by an intermediate variable. This variable is written in the output file that contains the customized model. The elements that do not contain any operation are kept without modification. We propagate the matrix obtained in the subsequent equations. Consequently, customizing eliminates multiplications by one and zero, and additions with zero. Moreover, if the robot has two or more successive revolute joints with parallel axes, it is more interesting to replace the corresponding product of matrices by a single matrix, which is calculated using equation [3.5]. We can also compute ${}^0\mathbf{s}_n$ using the vector product (${}^0\mathbf{n}_n \times {}^0\mathbf{a}_n$). In this case, the multiplication of the transformation matrices from the end-effector to the base saves the computation of the vectors $\mathbf{j}\mathbf{s}_n$ of the intermediate matrices ${}^j\mathbf{T}_n$, ($j = n, \dots, 1$).

• **Example 3.4.** Direct geometric model of the Stäubli RX-90 robot using the customized symbolic method.

a) computation of all the elements (s, n, a, P)

We denote T_{ijrs} as the element (r,s) of the matrix ${}^i\mathbf{T}_j$. As in Example 3.3, the product of the matrices is carried out starting from the last transformation matrix. We obtain the following intermediate variables for the matrix ${}^4\mathbf{T}_6$:

$$\begin{aligned} T_{4611} &= C_5 C_6 \\ T_{4612} &= -C_5 S_6 \\ T_{4631} &= S_5 C_6 \\ T_{4632} &= -S_5 S_6 \end{aligned}$$

Proceeding in the same way, the other intermediate variables are written as:

$$\begin{aligned} T_{3611} &= C_4 T_{4611} - S_4 S_6 \\ T_{3612} &= C_4 T_{4612} - S_4 C_6 \\ T_{3613} &= -C_4 S_5 \\ T_{3631} &= -S_4 T_{4611} - C_4 S_6 \\ T_{3632} &= -S_4 T_{4612} - C_4 C_6 \\ T_{3633} &= S_4 S_5 \\ T_{1314} &= D_3 C_2 \\ T_{1334} &= D_3 S_2 \\ T_{1611} &= C_{23} T_{3611} - S_{23} T_{4631} \\ T_{1612} &= C_{23} T_{3612} - S_{23} T_{4632} \\ T_{1613} &= C_{23} T_{3613} - S_{23} C_5 \\ T_{1614} &= -S_{23} R_{L4} + T_{1314} \\ T_{1631} &= S_{23} T_{3611} + C_{23} T_{4631} \\ T_{1632} &= S_{23} T_{3612} + C_{23} T_{4632} \end{aligned}$$

$$\begin{aligned} T1633 &= S23 T3613 + C23 C5 \\ T1634 &= C23 RL4 + T1334 \\ T0611 &= C1 T1611 + S1 T3631 \\ T0612 &= C1 T1612 + S1 T3632 \\ T0613 &= C1 T1613 + S1 T3633 \\ T0614 &= C1 T1614 \\ T0621 &= S1 T1611 - C1 T3631 \\ T0622 &= S1 T1612 - C1 T3632 \\ T0623 &= S1 T1613 - C1 T3633 \\ T0624 &= S1 T1614 \\ T0631 &= T1631 \\ T0632 &= T1632 \\ T0633 &= T1633 \\ T0634 &= T1634 \end{aligned}$$

Total number of operations: 44 multiplications and 18 additions

b) computing only the columns (\mathbf{n} , \mathbf{a} , \mathbf{P})

$$\begin{aligned}
T_{4612} &= -C_5 S_6 \\
T_{4632} &= -S_5 S_6 \\
T_{3612} &= C_4 T_{4612} - S_4 C_6 \\
T_{3613} &= -C_4 S_5 \\
T_{3632} &= -S_4 T_{4612} - C_4 C_6 \\
T_{3633} &= S_4 S_5 \\
T_{1314} &= D_3 C_2 \\
T_{1334} &= D_3 S_2 \\
T_{1612} &= C_{23} T_{3612} - S_{23} T_{4632} \\
T_{1613} &= C_{23} T_{3613} - S_{23} C_5 \\
T_{1614} &= -S_{23} R_{L4} + T_{1314} \\
T_{1632} &= S_{23} T_{3612} + C_{23} T_{4632} \\
T_{1633} &= S_{23} T_{3613} + C_{23} C_5 \\
T_{1634} &= C_{23} R_{L4} + T_{1334} \\
T_{0612} &= C_1 T_{1612} + S_1 T_{3632} \\
T_{0613} &= C_1 T_{1613} + S_1 T_{3633} \\
T_{0614} &= C_1 T_{1614} \\
T_{0622} &= S_1 T_{1612} - C_1 T_{3632} \\
T_{0623} &= S_1 T_{1613} - C_1 T_{3633} \\
T_{0624} &= S_1 T_{1614} \\
T_{0632} &= T_{1632} \\
T_{0633} &= T_{1633} \\
T_{0634} &= T_{1634}
\end{aligned}$$

Total number of operations: 30 multiplications and 12 additions

These equations constitute a complete direct geometric model. However, the computation of ${}^0\mathbf{s}_6$ requires six multiplications and three additions corresponding to the vector product (${}^0\mathbf{n}_6 \times {}^0\mathbf{a}_6$).

3.5. Transformation matrix of the end-effector in the world frame

The robot is a component among others in a robotic workcell. It is generally associated with fastening devices, sensors..., and eventually with other robots. Consequently, we have to define a reference world frame R_f , which may be different than the base reference frame R_0 of the robot (Figure 3.5). The transformation matrix defining R_0 with reference to R_f will be denoted as $\mathbf{Z} = {}^f\mathbf{T}_0$.

Moreover, very often, a robot is not intended to perform a single operation at the workcell: it has interchangeable different tools. In order to facilitate the programming of the task, it is more practical to define one or more functional frames, called *tool frames* for each tool. We denote $\mathbf{E} = {}^n\mathbf{T}_E$ as the transformation matrix defining the tool frame with respect to the terminal link frame.

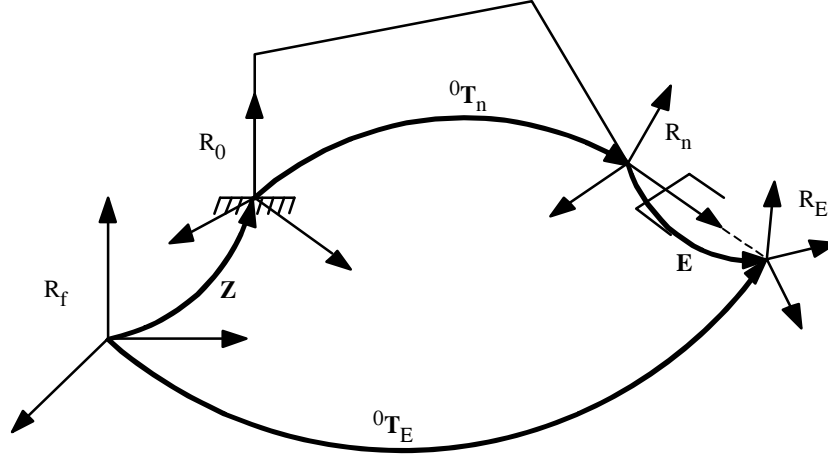


Figure 3.5. Transformations between the end-effector and the world frame

Thus, the transformation matrix fT_E can be written as:

$${}^fT_E = Z {}^0T_n(q) E \quad [3.13]$$

In most programming languages, the user can specify Z and E .

3.6. Specification of the orientation

Previously, we have used the elements of the matrix 0T_n to represent the position and orientation of the end-effector in frame R_0 . This means the use of the Cartesian coordinates to describe the position:

$${}^0P_n = [P_x \ P_y \ P_z]^T \quad [3.14]$$

and the use of the direction cosine matrix for the orientation:

$${}^0R_n = \begin{bmatrix} 0_{s_n} & 0_{n_n} & 0_{a_n} \end{bmatrix} \quad [3.15]$$

Practically, all the robot manufacturers make use of the Cartesian coordinates for the position even though the cylindrical or spherical representations could appear to be more judicious for some structures of robots.

Other representations may be used for the orientation, for example: Euler angles for CINCINNATI-T3 robots and PUMA robots, Roll-Pitch-Yaw (RPY) angles for

ACMA robots, Euler parameters for ABB robots. In this section, we will show how to obtain the direction cosines \mathbf{s} , \mathbf{n} , \mathbf{a} from the other representations and vice versa. Note that the orientation requires three independent parameters, thus the representation is redundant when it uses more than that.

3.6.1. Euler angles

The orientation of frame R_n expressed in frame R_0 is determined by specifying three angles, ϕ , θ and ψ , corresponding to a sequence of three rotations (Figure 3.6). The plane $(\mathbf{x}_n, \mathbf{y}_n)$ intersects the plane $(\mathbf{x}_0, \mathbf{y}_0)$ following the straight line ON, which is perpendicular to \mathbf{z}_0 and \mathbf{z}_n . The positive direction is given by the vector product $\mathbf{a}_0 \times \mathbf{a}_n$. The Euler angles are defined as:

- ϕ : angle between \mathbf{x}_0 and ON about \mathbf{z}_0 , with $0 \leq \phi < 2\pi$;
- θ : angle between \mathbf{z}_0 and \mathbf{z}_n about ON, with $0 \leq \theta \leq \pi$;
- ψ : angle between ON and \mathbf{x}_n about \mathbf{z}_n , with $0 \leq \psi < 2\pi$.

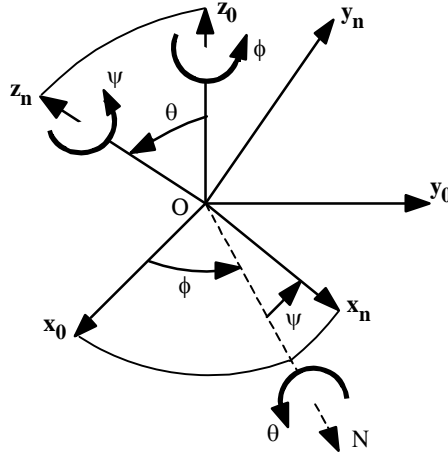


Figure 3.6. Euler angles (z, x, z representation)

The orientation matrix is given by:

$$\begin{aligned}
 {}^0R_n &= \text{rot}(\mathbf{z}, \phi) \text{rot}(\mathbf{x}, \theta) \text{rot}(\mathbf{z}, \psi) \\
 &= \begin{bmatrix} C\phi C\psi - S\phi C\theta S\psi & -C\phi S\psi - S\phi C\theta C\psi & S\phi S\theta \\ S\phi C\psi + C\phi C\theta S\psi & -S\phi S\psi + C\phi C\theta C\psi & -C\phi S\theta \\ S\theta S\psi & S\theta C\psi & C\theta \end{bmatrix} \quad [3.16]
 \end{aligned}$$

Inverse problem: expression of the Euler angles as functions of the direction cosines. Premultiplying equation [3.16] by $\mathbf{rot}(\mathbf{z}, -\phi)$, we obtain [Paul 81]:

$$\mathbf{rot}(\mathbf{z}, -\phi) {}^0\mathbf{R}_n = \mathbf{rot}(\mathbf{x}, \theta) \mathbf{rot}(\mathbf{z}, \psi) \quad [3.17]$$

Using relations [3.15] and [3.17] yields:

$$\begin{bmatrix} C\phi s_x + S\phi s_y & C\phi n_x + S\phi n_y & C\phi a_x + S\phi a_y \\ -S\phi s_x + C\phi s_y & -S\phi n_x + C\phi n_y & -S\phi a_x + C\phi a_y \\ s_z & n_z & a_z \end{bmatrix} = \begin{bmatrix} C\psi & -S\psi & 0 \\ C\theta S\psi & C\theta C\psi & -S\theta \\ S\theta S\psi & S\theta C\psi & C\theta \end{bmatrix} \quad [3.18]$$

Equating the (1, 3) elements of both sides, we obtain:

$$C\phi a_x + S\phi a_y = 0$$

which gives:

$$\begin{cases} \phi_1 = \text{atan2}(-a_x, a_y) \\ \phi_2 = \text{atan2}(a_x, -a_y) = \phi_1 + \pi \end{cases} \quad [3.19]$$

NOTE.— **atan2** is a mathematical function (Matlab, Fortran, ...), which provides the arc tangent function from its two arguments. This function has the following characteristics:

- examining the sign of both a_x and a_y allows us to uniquely determine the angle ϕ such that $-\pi \leq \phi < \pi$;
- the accuracy of this function is uniform over its full range of definition;
- when $a_x = 0$, $a_y = 0$, $a_z = \pm 1$ the angle ϕ is undefined (singularity).

Using the (2, 3) and (3, 3) elements of equation [3.18], we obtain:

$$\theta = \text{atan2}(S\phi a_x - C\phi a_y, a_z) \quad [3.20]$$

We proceed in a similar way to calculate ψ using the (1, 1) and (1, 2) elements:

$$\psi = \text{atan2}(-C\phi n_x - S\phi n_y, C\phi s_x + S\phi s_y) \quad [3.21]$$

When a_x and a_y are zero, the axes \mathbf{z}_n and \mathbf{z}_0 are aligned, thus θ is zero or π . This situation corresponds to the singular case: the rotations ϕ and ψ are about the same

axis and we can only determine their sum or difference. For example, when $a_z = 1$, we obtain:

$${}^0\mathbf{R}_n = \mathbf{rot}(\mathbf{z}, \psi + \phi)$$

and from this, we deduce:

$$\psi + \phi = \text{atan2}(-n_x, n_y) \quad [3.22]$$

NOTE.— The Euler angles adopted here correspond to a $(\mathbf{z}, \mathbf{x}, \mathbf{z})$ representation where the first rotation is about \mathbf{z}_0 , followed by a rotation about the new \mathbf{x} axis, followed by a last rotation about the new \mathbf{z} axis. Some authors prefer the $(\mathbf{z}, \mathbf{y}, \mathbf{z})$ representation [Paul 81]. A specific but interesting case can be encountered in the PUMA robot controller [Lee 83], [Dombre 88a] where an initial shift is introduced so that the orientation matrix is written as:

$${}^0\mathbf{R}_n = \mathbf{rot}(\mathbf{z}, \phi) \mathbf{rot}(\mathbf{x}, \theta + \frac{\pi}{2}) \mathbf{rot}(\mathbf{z}, \psi - \frac{\pi}{2}) \quad [3.23]$$

3.6.2. Roll-Pitch-Yaw angles

Following the convention shown in Figure 3.7, the angles ϕ , θ and ψ indicate roll, pitch and yaw respectively. If we suppose that the direction of motion (by analogy to the direction along which a ship is sailing) is along the \mathbf{z} axis, the orientation matrix can be written as:

$$\begin{aligned} {}^0\mathbf{R}_n &= \mathbf{rot}(\mathbf{z}, \phi) \mathbf{rot}(\mathbf{y}, \theta) \mathbf{rot}(\mathbf{x}, \psi) \\ &= \begin{bmatrix} C\phi C\theta & C\phi S\theta S\psi - S\phi C\psi & C\phi S\theta C\psi + S\phi S\psi \\ S\phi C\theta & S\phi S\theta S\psi + C\phi C\psi & S\phi S\theta C\psi - C\phi S\psi \\ -S\theta & C\theta S\psi & C\theta C\psi \end{bmatrix} \end{aligned} \quad [3.24]$$

Inverse problem: expression of the Roll-Pitch-Yaw angles as functions of the direction cosines. We use the same method discussed in the previous section. Premultiplying equation [3.24] by $\mathbf{rot}(\mathbf{z}, -\phi)$, we obtain:

$$\mathbf{rot}(\mathbf{z}, -\phi) {}^0\mathbf{R}_n = \mathbf{rot}(\mathbf{y}, \theta) \mathbf{rot}(\mathbf{x}, \psi) \quad [3.25]$$

which results in:

$$\begin{bmatrix} C\phi s_x + S\phi s_y & C\phi n_x + S\phi n_y & C\phi a_x + S\phi a_y \\ -S\phi s_x + C\phi s_y & -S\phi n_x + C\phi n_y & -S\phi a_x + C\phi a_y \\ s_z & n_z & a_z \end{bmatrix} = \begin{bmatrix} C\theta & S\theta S\psi & S\theta C\psi \\ 0 & C\psi & -S\psi \\ -S\theta & C\theta S\psi & C\theta C\psi \end{bmatrix} \quad [3.26]$$

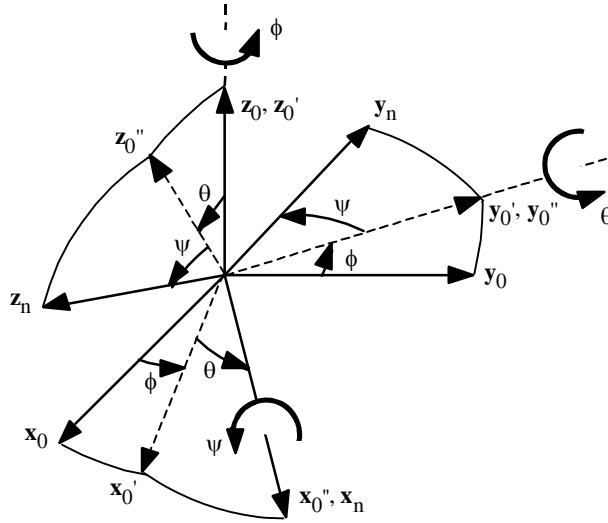


Figure 3.7. Roll-Pitch-Yaw angles

From the (2, 1) elements of equation [3.26], we obtain:

$$-S\phi s_x + C\phi s_y = 0$$

thus:

$$\begin{cases} \phi_1 = \text{atan2}(s_y, s_x) \\ \phi_2 = \text{atan2}(-s_y, -s_x) = \phi_1 + \pi \end{cases} \quad [3.27]$$

There is a singularity if s_x and s_y are zero ($\theta = \pm \frac{\pi}{2}$).

In the same way, from the (1, 1) and (1, 3) elements, then from the (2, 2) and (2, 3) elements, we deduce that:

$$\theta = \text{atan2}(-s_z, C\phi s_x + S\phi s_y) \quad [3.28]$$

$$\psi = \text{atan2}(S\phi a_x - C\phi a_y, -S\phi n_x + C\phi n_y) \quad [3.29]$$

3.6.3. Quaternions

The quaternions are also called *Euler parameters* or *Olinde-Rodrigues parameters*. In this representation, the orientation is expressed by four parameters that describe the orientation by a rotation of an angle θ ($0 \leq \theta \leq \pi$) about an axis of unit vector \mathbf{u} (Figure 3.8). We define the quaternions as:

$$\begin{cases} Q_1 = C(\theta/2) \\ Q_2 = u_x S(\theta/2) \\ Q_3 = u_y S(\theta/2) \\ Q_4 = u_z S(\theta/2) \end{cases} \quad [3.30]$$

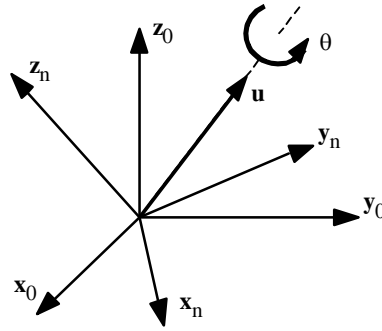


Figure 3.8. The quaternions

From these relations, we obtain:

$$Q_1^2 + Q_2^2 + Q_3^2 + Q_4^2 = 1 \quad [3.31]$$

The orientation matrix ${}^0\mathbf{R}_n$ is deduced from equation [2.30], defining $\mathbf{rot}(\mathbf{u}, \theta)$, after rewriting its elements as a function of Q_i . We note that:

$$C\theta = C^2(\theta/2) - S^2(\theta/2) = 2Q_1^2 - 1 \quad [3.32]$$

and that:

$$\left\{ \begin{array}{l}
 Q_2^2 = u_x^2 S^2 (\theta/2) = \frac{1}{2} u_x^2 (1 - C\theta); \\
 Q_3^2 = \frac{1}{2} u_y^2 (1 - C\theta); \\
 Q_4^2 = \frac{1}{2} u_z^2 (1 - C\theta); \\
 Q_2 Q_3 = \frac{1}{2} u_x u_y (1 - C\theta); \\
 Q_2 Q_4 = \frac{1}{2} u_x u_z (1 - C\theta); \\
 Q_3 Q_4 = \frac{1}{2} u_y u_z (1 - C\theta); \\
 u_x S\theta = 2u_x S(\theta/2)C(\theta/2) = 2Q_1 Q_2; \\
 u_y S\theta = 2Q_1 Q_3; \\
 u_z S\theta = 2Q_1 Q_4
 \end{array} \right. \quad [3.33]$$

Thus, the orientation matrix is given as:

$${}^0\mathbf{R}_n = \begin{bmatrix} 2(Q_1^2 + Q_2^2) - 1 & 2(Q_2 Q_3 - Q_1 Q_4) & 2(Q_2 Q_4 + Q_1 Q_3) \\ 2(Q_2 Q_3 + Q_1 Q_4) & 2(Q_1^2 + Q_3^2) - 1 & 2(Q_3 Q_4 - Q_1 Q_2) \\ 2(Q_2 Q_4 - Q_1 Q_3) & 2(Q_3 Q_4 + Q_1 Q_2) & 2(Q_1^2 + Q_4^2) - 1 \end{bmatrix} \quad [3.34]$$

For more information on the algebra of quaternions, the reader can refer to [de Casteljau 87].

Inverse problem: expression of the quaternions as functions of the direction cosines. Equating the elements of the diagonals of the right sides of equations [3.34] and [3.15] leads to:

$$Q_1 = \frac{1}{2} \sqrt{s_x + n_y + a_z + 1} \quad [3.35]$$

which is always positive. If we then subtract the (2, 2) and (3, 3) elements from the (1, 1) element, we can write after simplifying:

$$4 Q_2^2 = s_x - n_y - a_z + 1 \quad [3.36]$$

This expression gives the magnitude of Q_2 . For determining the sign, we consider the difference of the (3, 2) and (2, 3) elements, which leads to:

$$4 Q_1 Q_2 = n_z - a_y \quad [3.37]$$

The parameter Q_1 being always positive, the sign of Q_2 is that of $(n_z - a_y)$, which allows us to write:

$$Q_2 = \frac{1}{2} \text{sign}(n_z - a_y) \sqrt{s_x - n_y - a_z + 1} \quad [3.38]$$

Similar reasoning for Q_3 and Q_4 gives:

$$Q_3 = \frac{1}{2} \text{sign}(a_x - s_z) \sqrt{-s_x + n_y - a_z + 1} \quad [3.39]$$

$$Q_4 = \frac{1}{2} \text{sign}(s_y - n_x) \sqrt{-s_x - n_y + a_z + 1} \quad [3.40]$$

These expressions exhibit no singularity.

3.7. Conclusion

In this chapter, we have shown how to calculate the direct geometric model of a serial robot. This model is unique and is given in the form of explicit equations. The description of the geometry is based on rules that have an intrinsic logic facilitating its application. This method can be generalized to tree and closed loop structures (Chapter 7). It can also be extended to systems with lumped elasticity [Khalil 00a].

We have also presented the methods that are frequently used in robotics to specify the orientation of a body in space. We have shown how to calculate the orientation matrix from these representations and inversely, how to find the parameters of these descriptions from the orientation matrix.

Having calculated the direct geometric model, in the next chapter we study the inverse geometric problem, which consists of computing the joint variables as functions of a given location of the end-effector.

Chapter 4

Inverse geometric model of serial robots

4.1. Introduction

The direct geometric model of a robot provides the location of the end-effector in terms of the joint coordinates. The problem of computing the joint variables corresponding to a specified location of the end-effector is called the inverse geometric problem. This problem is at the center of computer control algorithms for robots. It has in general a multiple solution and its complexity is highly dependent on the geometry of the robot. The model that gives all the possible solutions for this problem is called the *Inverse Geometric Model* (IGM). In this chapter, we will present three methods to obtain the IGM of serial robots. First, we present the Paul method [Paul 81], which can be used to obtain an explicit solution for robots with relatively simple geometry that have many zero distances and parallel or perpendicular joint axes. Then, we develop a variation on the Pieper method [Pieper 68], which provides the analytical solution for the IGM of six degree-of-freedom robots with three prismatic joints or three revolute joints whose axes intersect at a point. Finally, we expose the Raghavan-Roth method [Raghavan 90], which gives the IGM for six degree-of-freedom robots with general (arbitrary) geometry using, at most, a sixteen degree polynomial.

When the inverse geometric model cannot be obtained or if it is difficult to implement in real time applications, iterative numerical techniques can be used. For this purpose, several algorithms can be found in the literature [Grudić 93]. Most of these algorithms use either the Newton-Raphson-based method [Pieper 68], [Goldenberg 85] or inverse Jacobian-based methods [Pieper 68], [Whitney 69], [Fournier 80], [Featherstone 83a]. In Chapter 6, we will present the second technique.

4.2. Mathematical statement of the problem

Let ${}^f\mathbf{T}_E^d$ be the homogeneous transformation matrix representing the desired location of the tool frame R_E relative to the world frame. In general, we can express ${}^f\mathbf{T}_E^d$ in the following form (§ 3.5):

$${}^f\mathbf{T}_E^d = \mathbf{Z} {}^0\mathbf{T}_n(\mathbf{q}) \mathbf{E} \quad [4.1]$$

where (Figure 3.5):

- \mathbf{Z} is the transformation matrix defining the location of the robot (frame R_0) relative to the world frame;
- ${}^0\mathbf{T}_n$ is the transformation matrix of the terminal frame R_n relative to frame R_0 . It is a function of the joint variable vector \mathbf{q} ;
- \mathbf{E} is the transformation matrix defining the tool frame R_E relative to R_n .

Putting all the known matrices of relation [4.1] on the left side leads to:

$$\mathbf{U}_{0=} {}^0\mathbf{T}_n(\mathbf{q}) \quad [4.2]$$

with $\mathbf{U}_0^d = \mathbf{Z}^{-1} {}^f\mathbf{T}_E^d \mathbf{E}^{-1}$

The problem is composed of a set of twelve nonlinear equations of n unknowns. The regular case has a finite number of solutions, whereas for redundant robots or in some singular configurations we obtain an infinite number of solutions. When the desired location is outside the reachable workspace there is no solution.

We say that a robot manipulator is *solvable* [Pieper 68], [Roth 76] when it is possible to compute all the joint configurations corresponding to a given location of the end-effector. Now, all non-redundant manipulators can be considered to be solvable [Lee 88], [Raghavan 90]. The number of solutions depends on the architecture of the robot manipulator and the amplitude of the joint limits. For six degree-of-freedom robots with only revolute joints (6R), or having five revolute joints and one prismatic joint (5R1P), the maximum number of solutions is sixteen. When the robot has three revolute joints whose axes intersect at a point, the maximum number of solutions is eight. For the 3P3R robots, this number reduces to two. In all cases, it decreases when the geometric parameters take certain particular values.

Robots with less than six degrees of freedom are not able to place the end-effector frame in an arbitrary location. Thus, we only specify the task in terms of placing some elements of the tool frame (points, axes) in the world frame. Under

these conditions, the matrix \mathbf{E} is not completely defined, and the equation to solve is given by:

$$\mathbf{Z}^{-1} \mathbf{f} \mathbf{T}_E^d = {}^0 \mathbf{T}_n(\mathbf{q}) \mathbf{E} \quad [4.3]$$

4.3. Inverse geometric model of robots with simple geometry

For robots with simple geometry, where most of the distances (r_j and d_j) are zero and most of the angles (θ_j and α_j) are zero or $\pm \pi/2$, the inverse geometric model can be analytically obtained using the Paul method [Paul 81]. Most commercially available robots can be solved using this method.

4.3.1. Principle

Let us consider a robot manipulator whose transformation matrix has the expression:

$${}^0 \mathbf{T}_n = {}^0 \mathbf{T}_1(q_1) {}^1 \mathbf{T}_2(q_2) \dots {}^{n-1} \mathbf{T}_n(q_n) \quad [4.4]$$

Let \mathbf{U}_0^d be the desired location such that:

$$\mathbf{U}_0^d = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [4.5]$$

The IGM is obtained by solving the following equation:

$$\mathbf{U}_0^d = {}^0 \mathbf{T}_1(q_1) {}^1 \mathbf{T}_2(q_2) \dots {}^{n-1} \mathbf{T}_n(q_n) \quad [4.6]$$

To find the solutions of this equation, Paul [Paul 81] proposed to move each joint variable to the left side one after the other by successively premultiplying equation [4.6] by ${}^j \mathbf{T}_{j-1}$, for j varying from 1 to $n-1$. Then, the joint variables are determined by equating the elements of the two sides of each equation. For example, for a six degree-of-freedom robot, we proceed as follows:

- premultiply equation [4.6] by ${}^1 \mathbf{T}_0$:

$${}^1\mathbf{T}_0\mathbf{U}_0^d = {}^1\mathbf{T}_2{}^2\mathbf{T}_3{}^3\mathbf{T}_4{}^4\mathbf{T}_5{}^5\mathbf{T}_6 \quad [4.7]$$

The elements of the left side are constants or functions of q_1 . The elements of the right side are constants or functions of q_2, \dots, q_6 ;

- try to solve q_1 by equating the elements of the two sides of equation [4.7];
- premultiply equation [4.7] by ${}^2\mathbf{T}_1$ and try to determine q_2 ;
- continue the process until all the variables are solved.

In summary, the equations used to obtain all the joint variables are written as:

$$\begin{aligned} {}^1\mathbf{T}_0\mathbf{U}_0^d &= {}^1\mathbf{T}_2{}^2\mathbf{T}_3{}^3\mathbf{T}_4{}^4\mathbf{T}_5{}^5\mathbf{T}_6 \\ {}^2\mathbf{T}_1{}^1\mathbf{T}_0\mathbf{U}_0^d &= {}^2\mathbf{T}_3{}^3\mathbf{T}_4{}^4\mathbf{T}_5{}^5\mathbf{T}_6 \\ {}^3\mathbf{T}_2{}^2\mathbf{T}_0\mathbf{U}_0^d &= {}^3\mathbf{T}_4{}^4\mathbf{T}_5{}^5\mathbf{T}_6 \\ {}^4\mathbf{T}_3{}^3\mathbf{T}_0\mathbf{U}_0^d &= {}^4\mathbf{T}_5{}^5\mathbf{T}_6 \\ {}^5\mathbf{T}_4{}^4\mathbf{T}_0\mathbf{U}_0^d &= {}^5\mathbf{T}_6 \\ \text{with } \mathbf{U}_j &= {}^j\mathbf{T}_6 = {}^j\mathbf{T}_{j-1} \mathbf{U}_{j-1} \end{aligned} \quad [4.8]$$

The resolution of equations [4.8] needs intuition, but the use of this method on a large number of industrial robots has shown that only few fundamental types of equations are encountered [Khalil 86b] (Table 4.1). The solutions of these equations are given in Appendix 1.

NOTES.—

- the matrices of the right side of equations [4.8] are already available when computing the direct geometric model (DGM) if the multiplication of the transformation matrices is started from the end of the robot;
- in certain cases, it may be more convenient to solve the robot by first determining q_n and ending with q_1 . In this case, we postmultiply equation [4.6] by ${}^j\mathbf{T}_{j-1}$ for j varying from n to 2.

Table 4.1. Types of equations encountered in the Paul method

Type 1	$X r_i = Y$
Type 2	$X S\theta_i + Y C\theta_i = Z$
Type 3	$X_1 S\theta_i + Y_1 C\theta_i = Z_1$ $X_2 S\theta_i + Y_2 C\theta_i = Z_2$
Type 4	$X_1 r_j S\theta_i = Y_1$ $X_2 r_j C\theta_i = Y_2$
Type 5	$X_1 S\theta_i = Y_1 + Z_1 r_j$ $X_2 C\theta_i = Y_2 + Z_2 r_j$
Type 6	$W S\theta_j = X C\theta_i + Y S\theta_i + Z_1$ $W C\theta_j = X S\theta_i - Y C\theta_i + Z_2$
Type 7	$W_1 C\theta_j + W_2 S\theta_j = X C\theta_i + Y S\theta_i + Z_1$ $W_1 S\theta_j - W_2 C\theta_j = X S\theta_i - Y C\theta_i + Z_2$
Type 8	$X C\theta_i + Y C(\theta_i + \theta_j) = Z_1$ $X S\theta_i + Y S(\theta_i + \theta_j) = Z_2$

r_j : prismatic joint variable,

$S\theta_i, C\theta_i$: sine and cosine of a revolute joint variable θ_i .

4.3.2. Special case: robots with a spherical wrist

Most six degree-of-freedom industrial robots have a spherical wrist composed of three revolute joints whose axes intersect at a point (Figure 4.1). This structure is characterized by the following set of geometric parameters:

$$\begin{cases} d_5 = r_5 = d_6 = 0 \\ \sigma_4 = \sigma_5 = \sigma_6 = 0 \\ S\alpha_5 \neq 0, S\alpha_6 \neq 0 \quad (\text{non-redundant robot}) \end{cases}$$

The position of the center of the spherical joint is obtained as a function of the joint variables q_1, q_2 and q_3 . This type of structure allows the decomposition of the six degree-of-freedom problem into two equations each with three unknowns. They represent a position equation and an orientation equation. The position problem, which is a function of q_1, q_2 and q_3 , is first solved, then the orientation problem allows us to determine $\theta_4, \theta_5, \theta_6$.

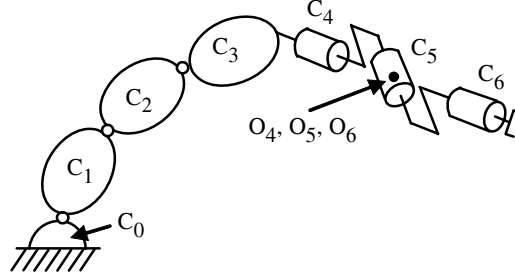


Figure 4.1. Six degree-of-freedom robot with a spherical wrist

4.3.2.1. Position equation

Since ${}^0\mathbf{P}_6 = {}^0\mathbf{P}_4$, the fourth column of the transformation matrix ${}^0\mathbf{T}_4$ is equal to the fourth column of \mathbf{U}_0^d :

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 {}^3\mathbf{T}_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad [4.9]$$

We obtain the variables q_1, q_2, q_3 by successively premultiplying this equation by ${}^j\mathbf{T}_0$, $j = 1, 2$, to isolate and determine sequentially the joint variables. The elements of the right side have already been calculated for the DGM.

4.3.2.2. Orientation equation

The orientation part of equation [4.2] is written as:

$$[\mathbf{s} \quad \mathbf{n} \quad \mathbf{a}] = {}^0\mathbf{R}_6(\mathbf{q})$$

yielding:

$${}^3\mathbf{R}_0(q_1, q_2, q_3)[\mathbf{s} \quad \mathbf{n} \quad \mathbf{a}] = {}^3\mathbf{R}_6(\theta_4, \theta_5, \theta_6)$$

which can be written as:

$$[\mathbf{F} \quad \mathbf{G} \quad \mathbf{H}] = {}^3\mathbf{R}_6(\theta_4, \theta_5, \theta_6) \quad [4.10]$$

Since q_1, q_2, q_3 have been determined, the left side elements are considered to be known. To obtain $\theta_4, \theta_5, \theta_6$, we successively premultiply equation [4.10] by ${}^4\mathbf{R}_3$ then by ${}^5\mathbf{R}_4$ and proceed by equating the elements of the two sides. Again, the elements of the right side have already been calculated for the DGM.

• **Example 4.1.** IGM of the Stäubli RX-90 robot. The geometric parameters are given in Table 3.1. The robot has a spherical wrist. The DGM is developed in Chapter 3.

a) Computation of $\theta_1, \theta_2, \theta_3$

i) by developing equation [4.9], we obtain:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} C1(-S23RL4 + C2D3) \\ S1(-S23RL4 + C2D3) \\ C23RL4 + S2D3 \\ 1 \end{bmatrix}$$

Note that the elements of the right side constitute the fourth column of ${}^0\mathbf{T}_6$, which have already been calculated for the DGM. No variable can be determined from this equation;

ii) premultiplying the previous equation by ${}^1\mathbf{T}_0$, we obtain the left side elements as:

$$U(1) = C1 P_x + S1 P_y$$

$$U(2) = -S1 P_x + C1 P_y$$

$$U(3) = P_z$$

The elements of the right side are obtained from the fourth column of ${}^1\mathbf{T}_6$:

$$T(1) = -S23 RL4 + C2 D3$$

$$T(2) = 0$$

$$T(3) = C23 RL4 + S2 D3$$

By equating $U(2)$ and $T(2)$, we obtain the following two solutions for θ_1 :

$$\begin{cases} \theta_{1,1} = \text{atan2}(P_y, P_x) \\ \theta_{1,2} = \theta_{1,1} + \pi \end{cases}$$

iii) premultiplying by ${}^2\mathbf{T}_1$, we obtain the elements of the left side as:

$$U(1) = C2 (C1 P_x + S1 P_y) + S2 P_z$$

$$U(2) = -S2 (C1 P_x + S1 P_y) + C2 P_z$$

$$U(3) = S1 P_x - C1 P_y$$

The elements of the right side represent the fourth column of ${}^2\mathbf{T}_6$:

$$T(1) = -S3 RL4 + D3$$

$$T(2) = C3 RL4$$

$$T(3) = 0$$

We determine θ_2 and θ_3 by considering the first two elements, which constitute a type-6 system of equations (Table 4.1). First, an equation in θ_2 is obtained:

$$X S2 + Y C2 = Z$$

with:

$$X = -2P_z D3$$

$$Y = -2 B_1 D3$$

$$Z = RL_4^2 - D_3^2 - P_z^2 - B_1^2$$

$$B_1 = P_x C1 + P_y S1$$

from which we deduce that:

$$C2 = \frac{YZ - \varepsilon X \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2}$$

$$S2 = \frac{XZ + \varepsilon Y \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2}$$

$$\text{with } \varepsilon = \pm 1$$

This gives two solutions of the following form:

$$\theta_2 = \text{atan2}(S2, C2)$$

θ_2 being known, we obtain:

$$\theta_3 = \text{atan2}(S3, C3)$$

with:

$$\begin{cases} S3 = \frac{-PzS2 - B_1C2 + D3}{RL4} \\ C3 = \frac{-B_1S2 + PzC2}{RL4} \end{cases}$$

b) Computation of $\theta_4, \theta_5, \theta_6$

Once the variables $\theta_1, \theta_2, \theta_3$ are determined, we define the (3×3) orientation matrix ${}^3\mathbf{R}_6$ as follows:

$$[\mathbf{F} \quad \mathbf{G} \quad \mathbf{H}] = {}^3\mathbf{R}_0 [\mathbf{s} \quad \mathbf{n} \quad \mathbf{a}]$$

The elements of \mathbf{F} are written as:

$$U(1,1) = C23 (C1 s_x + S1 s_y) + S23 s_z$$

$$U(2,1) = -S23 (C1 s_x + S1 s_y) + C23 s_z$$

$$U(3,1) = S1 s_x - C1 s_y$$

The elements of \mathbf{G} and \mathbf{H} are obtained from \mathbf{F} by replacing (s_x, s_y, s_z) by (n_x, n_y, n_z) and (a_x, a_y, a_z) respectively.

i) equating the elements of $[\mathbf{F} \quad \mathbf{G} \quad \mathbf{H}] = {}^3\mathbf{R}_6$

The elements of ${}^3\mathbf{R}_6$ are obtained from ${}^3\mathbf{T}_6$, which is calculated for the DGM:

$${}^3\mathbf{R}_6 = \begin{bmatrix} C6C5C4 - S6S4 & -S6C5C4 - C6S4 & -S5C4 \\ C6S5 & -S6S5 & C5 \\ -C6C5S4 - S6C4 & S6C5S4 - C6C4 & S5S4 \end{bmatrix}$$

We can determine θ_5 from the (2, 3) elements using an arccosine function. But this solution is not retained, considering that another one using an atan2 function may appear in the next equation s;

ii) equating the elements of ${}^4\mathbf{R}_3[\mathbf{F} \ \mathbf{G} \ \mathbf{H}] = {}^4\mathbf{R}_6$

The elements of the first column of the left side are written as:

$$U(1, 1) = C4 F_x - S4 F_z$$

$$U(2, 1) = -C4 F_z - S4 F_x$$

$$U(3, 1) = F_y$$

The elements of the second and third columns are obtained by replacing (F_x, F_y, F_z) with (G_x, G_y, G_z) and (H_x, H_y, H_z) respectively. The elements of ${}^4\mathbf{R}_6$ are obtained from ${}^4\mathbf{T}_6$, which has already been calculated for the DGM:

$${}^4\mathbf{R}_6 = \begin{bmatrix} C6C5 & -S6C5 & -S5 \\ S6 & C6 & 0 \\ C6S5 & -S6S5 & C5 \end{bmatrix}$$

From the (2, 3) elements, we obtain a type-2 equation in θ_4 :

$$-C4 H_z - S4 H_x = 0$$

which gives two solutions:

$$\begin{cases} \theta_{4,1} = \text{atan2}(H_z, -H_x) \\ \theta_{4,2} = \theta_{4,1} + \pi \end{cases}$$

From the (1, 3) and (3, 3) elements, we obtain a type-3 system of equations in θ_5 :

$$-S5 = C4 H_x - S4 H_z$$

$$C5 = H_y$$

whose solution is:

$$\theta_5 = \text{atan2}(S5, C5)$$

Finally, by considering the (2, 1) and (2, 2) elements, we obtain a type-3 system of equations in θ_6 :

$$\begin{aligned} S_6 &= -C_4 F_z - S_4 F_x \\ C_6 &= -C_4 G_z - S_4 G_x \end{aligned}$$

whose solution is:

$$\theta_6 = \text{atan2}(S_6, C_6)$$

NOTES.– By examining the IGM solution of the Stäubli RX-90 robot, it can be observed that:

a) Number of solutions: in the regular case, the Stäubli RX-90 robot has eight solutions for the IGM (product of the number of possible solutions for each joint). Some of these configurations may not be accessible because of the joint limits.

a) The robot has the following singular positions:

- i) *shoulder singularity*: takes place when the point O_6 lies on the z_0 axis (Figure 4.2a). Thus $P_x = P_y = 0$, which corresponds to $S_2 S_3 R_4 - C_2 D_3 = 0$. In this case, both the two arguments of the atan2 function used to determine θ_1 are zero, thus leaving θ_1 undetermined. We are free to choose any value for θ_1 , but frequently the current value is assigned. This means that one can always find a solution, but when leaving this configuration, a small change in the desired location may require a significant variation in θ_1 , impossible to realize due to the speed and acceleration limits of the actuator;
- ii) *wrist singularity*: takes place when $C_2 S_3 (C_1 a_x + S_1 a_y) + S_2 S_3 a_z = H_x = 0$ and $(S_1 a_x - C_1 a_y) = H_z = 0$. The two arguments of the atan2 function used to determine θ_4 are zero. From the (2, 3) element of 3R_6 , we notice that in this case $C\theta_5 = \pm 1$. Thus, the axes of joints 4 and 6 are collinear and it is the sum $\theta_4 \pm \theta_6$ that can be determined (Figure 4.2b). For example, when $\theta_5 = 0$, the orientation equation becomes:

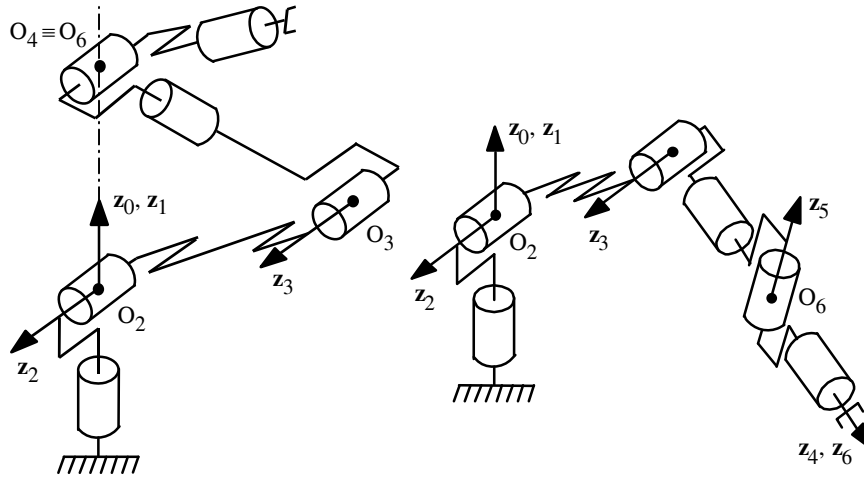
$$[\mathbf{F} \quad \mathbf{G} \quad \mathbf{H}] = {}^3R_6 = \begin{bmatrix} C_4 C_6 & -S_4 C_6 & 0 \\ 0 & 0 & 1 \\ -S_4 C_6 & -C_4 C_6 & 0 \end{bmatrix}$$

Thus, $\theta_4 + \theta_6 = \text{atan2}(-G_x, -G_z)$. We can arbitrarily assign θ_4 to its current value and calculate the corresponding θ_6 . We can also calculate the values of θ_4 and θ_6 for which the joints 4 and 6 move away from their limits;

iii) *elbow singularity*: occurs when $C3 = 0$. This singularity will be discussed in Chapter 6. It does not affect the inverse geometric model computation (Figure 4.2c).

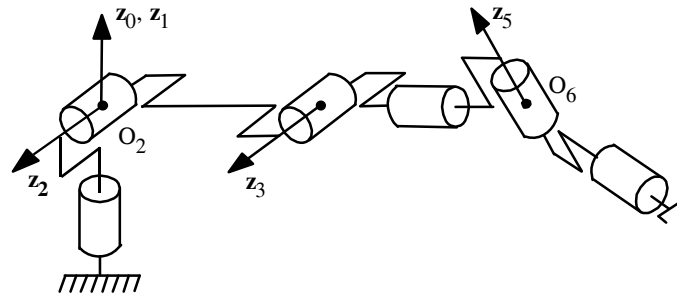
b) The above-mentioned singularities are classified as first order singularities. Singularities of higher order may occur when several singularities of first order take place simultaneously.

c) Number of solutions: in the regular case, the Stäubli RX-90 robot has eight solutions for the IGM (product of the number of possible solutions for each joint). Some of these configurations may not be accessible because of the joint limits.



a) Shoulder singularity
($P_x = P_y = 0$ or $S23RL4 - C2D3 = 0$)

b) Wrist singularity ($S5 = 0$)



c) Elbow singularity ($C3 = 0$)

Figure 4.2. Singular positions of the Stäubli RX-90 robot**4.3.3. Inverse geometric model of robots with more than six degrees of freedom**

A robot with more than six degrees of freedom is redundant and its inverse geometric problem has an infinite number of solutions. To obtain a closed form solution, $(n-6)$ additional relations are needed. Two strategies are possible:

- arbitrarily fixing $(n-6)$ joint values to reduce the problem to six unknowns. The selection of the fixed joints is determined by the task specifications and the robot structure;
- introducing $(n-6)$ additional relations describing the redundancy, as is done in certain seven degree-of-freedom robots [Hollerbach 84b].

4.3.4. Inverse geometric model of robots with less than six degrees of freedom

When the robot has less than six degrees of freedom, the end-effector frame R_E cannot be placed at an arbitrary location except if certain elements of ${}^0T_E^d$ have specific values to compensate for the missing degrees of freedom. Otherwise, instead of realizing frame-to-frame task, we consider tasks with less degrees of freedom such as point-to-point contact, or (point-axis) to (point-axis) contact [Manaoui 85].

In the next example, we will study this problem for the four degree-of-freedom SCARA robot whose geometric parameters are given in Table 3.2.

• **Example 4.2.** IGM of the SCARA robot (Figure 3.4).

i) frame-to-frame contact

In this case, the system of equations to be solved is given by equation [4.2] and U_0 is defined by equation [4.5]:

$$U_0^d = {}^0T_4 = \begin{bmatrix} C123 & -S123 & 0 & C12D3 + C1D2 \\ S123 & C123 & 0 & S12D3 + S1D2 \\ 0 & 0 & 1 & r_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Examining the elements of this matrix reveals that frame-to-frame task is possible if the third column of the desired U_0 is equal to $[0 \ 0 \ 1 \ 0]^T$. This implies two independent conditions, which compensate for the missing degrees of freedom. By equating the (3, 4) elements, we obtain:

$$r_4 = P_z$$

The (1, 4) and (2, 4) elements give a type-8 system of equations in θ_1 and θ_2 with the following solution:

$$\begin{aligned}\theta_2 &= \arctan 2(\pm\sqrt{1-C2^2}, C2) \\ \theta_1 &= \arctan 2(S1, C1)\end{aligned}$$

with:

$$\begin{aligned}C2 &= \frac{D^2 - D_2^2 - D_3^2}{2D_2D_3} \\ D^2 &= P_x^2 + P_y^2 \\ S1 &= \frac{B1P_y - B2P_x}{D^2} \quad C1 = \frac{B1P_x + B2P_y}{D^2}\end{aligned}$$

$$B1 = D2 + D3 C2, \quad B2 = D3 S2$$

After determining θ_1 and θ_2 , we obtain θ_3 as:

$$\theta_3 = \arctan 2(s_y, s_x) - \theta_2 - \theta_1$$

ii) (point-axis) to (point-axis) contact

Let us suppose that the tool is defined by an axis of unit vector \mathbf{a}_E , passing by O_E such that:

$${}^4\mathbf{P}_E = [Q_x \quad Q_y \quad Q_z]^T$$

$${}^4\mathbf{a}_E = [W_x \quad W_y \quad W_z]^T$$

The task consists of placing the point O_E at a point of the environment while aligning the axis \mathbf{a}_E with an axis of the environment, which are defined by:

$${}^0\mathbf{P}_E^d = [P_x \quad P_y \quad P_z]^T$$

$${}^0\mathbf{a}_E^d = [a_x \quad a_y \quad a_z]^T$$

The system to be solved is written as:

$$\begin{bmatrix} - & - & a_x & P_x \\ - & - & a_y & P_y \\ - & - & a_z & P_z \\ - & - & 0 & 1 \end{bmatrix} = {}^0T_4 \begin{bmatrix} - & - & W_x & Q_x \\ - & - & W_y & Q_y \\ - & - & W_z & Q_z \\ - & - & 0 & 1 \end{bmatrix}$$

After simplifying, we obtain:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} Q_x C_{123} - Q_y S_{123} + C_{12} D_3 + C_{12} D_2 \\ Q_x S_{123} + Q_y C_{123} + S_{12} D_3 + S_{12} D_2 \\ Q_z + r_4 \end{bmatrix}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} W_x C_{123} - W_y S_{123} \\ W_x S_{123} + W_y C_{123} \\ W_z \end{bmatrix}$$

Thus, we deduce that the condition $a_z = W_z$ must be satisfied to realize the task. The IGM solutions are obtained in the following way:

- from the a_x and a_y equations, we obtain $(\theta_1 + \theta_2 + \theta_3)$ by solving a type-3 system (Appendix 1):

$$\theta_1 + \theta_2 + \theta_3 = \text{atan2}(S_{123}, C_{123})$$

$$\text{with } S_{123} = \frac{a_y W_x - a_x W_y}{W_x^2 + W_y^2} \text{ and } C_{123} = \frac{a_x W_x + a_y W_y}{W_x^2 + W_y^2}$$

with $(W_x^2 + W_y^2 \neq 0)$;

- when $W_x = W_y = 0$, the axis of the end-effector is vertical and its orientation cannot be changed. Any value for θ_3 may be taken;
- from P_x and P_y equations, we obtain θ_1 and θ_2 by solving a type-8 system of equations;
- finally, from the third element of the position equation, we obtain $r_4 = P_z - Q_z$.

In summary, the task of a SCARA robot can be described in one of the following ways:

- placing the tool frame onto a specified frame provided that the third column of the matrix ${}^0\mathbf{T}_4^d = {}^0\mathbf{T}_E^d \mathbf{E}^{-1} = [0 \ 0 \ 1 \ 0]^T$, in order to satisfy that \mathbf{z}_4 is vertical;
- placing an axis and a point of the tool frame respectively onto an axis and a point of the environment provided that $a_z = W_z$. The obvious particular case is to locate a horizontal axis of the end-effector frame in a horizontal plane ($a_z = W_z = 0$).

4.4. Inverse geometric model of decoupled six degree-of-freedom robots

4.4.1. Introduction

The IGM of a six degree-of-freedom decoupled robot can be computed by solving two sub-problems, each having three unknowns [Pieper 68]. Two classes of structures are considered:

- a) robots having a spherical joint given by one of the following four combinations: XXX(RRR), (RRR)XXX, XX(RRR)X, X(RRR)XX, where (RRR) denotes a spherical joint and X denotes either a revolute (R) or a prismatic (P) joint. Consequently, each combination results in eight structures;
- b) robots having three revolute and three prismatic joints as given by one of the following 20 combinations: PPPRRR, PPRPRR, PPRRPR, ...

In this section, we present the inverse geometric model of these structures using two general equations [Khalil 90c], [Bennis 91a]. These equations make use of the six types of equations shown in Table 4.2. The first three types have already been used in the Paul method (Table 4.1). The explicit solution of a type-10 equation can be obtained symbolically using software packages like Maple or Mathematica. In general, however, the numerical solution is more accurate. We note that a type-11 equation can be transformed into type-10 using the half-angle transformation by writing $C\theta_i$ and $S\theta_i$ as:

$$C\theta_i = \frac{1-t^2}{1+t^2} \quad \text{and} \quad S\theta_i = \frac{2t}{1+t^2} \quad \text{with} \quad t = \tan \frac{\theta_i}{2}$$

Table 4.2. Types of equations encountered in the Pieper method

Type 1	$X r_i = Y$
Type 2	$X C\theta_i + Y S\theta_i = Z$
Type 3	$X_1 S\theta_i + Y_1 C\theta_i = Z_1$ $X_2 S\theta_i + Y_2 C\theta_i = Z_2$
Type 9	$a_2 r_i^2 + a_1 r_i + a_0 = 0$
Type 10	$a_4 r_i^4 + a_3 r_i^3 + a_2 r_i^2 + a_1 r_i + a_0 = 0$
Type 11	$a_4 S\theta_i^2 + a_3 C\theta_i S\theta_i + a_2 C\theta_i + a_1 S\theta_i + a_0 = 0$

4.4.2. Inverse geometric model of six degree-of-freedom robots having a spherical joint

In this case, equation [4.6] is decoupled into two equations, each containing three variables:

- a position equation, which is a function of the joint variables that do not belong to the spherical joint;
- an orientation equation, which is a function of the joint variables of the spherical joint.

4.4.2.1. General solution of the position equation

The revolute joint axes $m-1$, m and $m+1$ ($2 \leq m \leq 5$) form a spherical joint if:

$$\begin{cases} d_m = r_m = d_{m+1} = 0 \\ S\alpha_m \neq 0 \\ S\alpha_{m+1} \neq 0 \end{cases}$$

The position of the center of the spherical joint is represented as: O_{m-1} , O_{m-1} and O'_{m+1} , where O'_{m+1} is the translation of O_{m+1} by . It is independent of the joint variables θ_{m-1} , θ_m and θ_{m+1} . Thus, we can show (Figure 4.3) that the position of O_m relative to frame R_{m-2} is given by:

$${}^{m-2}\mathbf{T}_{m+1} \mathbf{Trans}(z, -r_{m+1}) \mathbf{p}_0 = \begin{bmatrix} {}^{m-2}\mathbf{p}_{m-1} \\ 1 \end{bmatrix} = \begin{bmatrix} d_{m-1} \\ -r_{m-1} S\alpha_{m-1} \\ r_{m-1} C\alpha_{m-1} \\ 1 \end{bmatrix} \quad [4.11]$$

where $\mathbf{p}_0 = [0 \ 0 \ 0 \ 1]^T$ and ${}^{m-2}\mathbf{p}_{m-1}$ is obtained using equation [3.2].

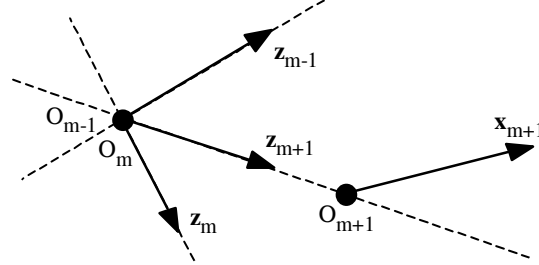


Figure 4.3. Axes of a spherical joint

To obtain the position equation, we write equation [4.6] in the following form:

$${}^0\mathbf{T}_{m-2} {}^{m-2}\mathbf{T}_{m+1} {}^{m+1}\mathbf{T}_6 = \mathbf{U}_0 \quad [4.12]$$

Postmultiplying this relation by ${}^6\mathbf{T}_{m+1} \mathbf{Trans}(\mathbf{z}, -\mathbf{r}_{m+1}) \mathbf{p}_0$ and using equation [4.11], we obtain:

$${}^0\mathbf{T}_{m-2} \begin{bmatrix} {}^{m-2}\mathbf{p}_{m-1} \\ 1 \end{bmatrix} = \mathbf{U}_0 {}^d\mathbf{T}_{m+1} \mathbf{Trans}(\mathbf{z}, -\mathbf{r}_{m+1}) \mathbf{p}_0 \quad [4.13]$$

Equation [4.13] can be written in the general form:

$$\mathbf{Rot}(\mathbf{z}, \theta_i) \mathbf{Trans}(\mathbf{z}, r_i) \mathbf{Rot}(\mathbf{x}, \alpha_j) \mathbf{Trans}(\mathbf{x}, d_j) \mathbf{Rot}(\mathbf{z}, \theta_j) \mathbf{Trans}(\mathbf{z}, r_j) \begin{bmatrix} \mathbf{f}(q_k) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ 1 \end{bmatrix} \quad [4.14]$$

where:

- the subscripts i, j and k represent the joints that do not belong to the spherical joint; i and j represent two successive joints;
- the vector \mathbf{f} is a function of the joint variable q_k ;
- the vector \mathbf{g} is a constant.

By combining the parameters \bar{q}_i and \bar{q}_j with \mathbf{g} and \mathbf{f} respectively, equation [4.14] becomes:

$$\mathbf{Rot/Trans}(\mathbf{z}, q_i) \mathbf{Rot}(\mathbf{x}, \alpha_j) \mathbf{Trans}(\mathbf{x}, d_j) \mathbf{Rot/Trans}(\mathbf{z}, q_j) \begin{bmatrix} \mathbf{F}(q_k) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ 1 \end{bmatrix} \quad [4.15]$$

with:

- $\mathbf{Rot/Trans}(z, q_i) = \mathbf{Rot}(z, \theta_i)$ if $q_i = \theta_i$
 $= \mathbf{Trans}(z, r_i)$ if $q_i = r_i$
- $\begin{bmatrix} \mathbf{F}(q_k) \\ 1 \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ 1 \end{bmatrix} = \mathbf{Rot/Trans}(z, \bar{q}_j) \begin{bmatrix} \mathbf{f}(q_k) \\ 1 \end{bmatrix}$
- $\begin{bmatrix} \mathbf{G} \\ 1 \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \\ G_z \\ 1 \end{bmatrix} = \mathbf{Rot/Trans}(z, -\bar{q}_i) \begin{bmatrix} \mathbf{g} \\ 1 \end{bmatrix}$
- $\mathbf{Rot/Trans}(z, \bar{q}_i) = \mathbf{Trans}(z, r_i)$ if joint i is revolute
 $= \mathbf{Rot}(z, \theta_i)$ if joint i is prismatic

The components of \mathbf{G} are constants and those of \mathbf{F} are functions of the joint variable q_k . We note that if joint k is revolute, then:

$$\|\mathbf{F}\|^2 = a C\theta_k + b S\theta_k + c \quad [4.16]$$

where a , b and c are constants.

Table 4.3 shows the equations that are used to obtain the joint variable q_k according to the types of joints i and j (columns 1 and 2). The variables q_i and q_j are then computed using equation [4.15]. Table 4.4 indicates the type of the obtained equations and the maximum number of solutions for each structure; the last column of the table indicates the order in which we calculate them. In Example 4.3, we will develop the solution for the case where joints i and j are revolute. We note that the maximum number of solutions for q_i , q_j and q_k is four.

NOTE.— The assignment of i , j and k for the joints that do not belong to the spherical joint is not unique. In order to get a simple solution for q_k , this assignment can be changed using the concept of the inverse robot (presented in Appendix 2). For instance, if the spherical joint is composed of the joints 4, 5 and 6, we can take $i = 1$, $j = 2$, $k = 3$. But we can also take $i = 3$, $j = 2$, $k = 1$ by using the concept of the inverse robot. We can easily verify that the second choice is more interesting if these joints are revolute and $S\alpha_2 \neq 0$, $d_2 \neq 0$ but $d_3 = 0$ or $S\alpha_3 = 0$.

Table 4.3. Solutions of q_k and types of equations

i	j	Conditions	Equations for q_k	Type	
				θ_k	r_k
R	R	$S\alpha_j = 0$	$C\alpha_j F_z(q_k) = G_z$	2	1
		$d_j = 0$	$\ F\ ^2 = \ G\ ^2$	2	9
		$d_j \neq 0$ and $S\alpha_j \neq 0$	$\left[\frac{\ F\ ^2 - \ G\ ^2 - d_j^2}{2d_j} \right]^2 + \left[\frac{F_z - C\alpha_j G_z}{S\alpha_j} \right]^2 = G_x^2 + G_y^2$	11	10
R	P	$C\alpha_j = 0$	$F_y(q_k) = S\alpha_j G_z$	2	1
		$C\alpha_j \neq 0$	$(F_x + d_j)^2 + \left[\frac{F_y - S\alpha_j G_z}{C\alpha_j} \right]^2 = G_x^2 + G_y^2$	11	9
P	R	$C\alpha_j = 0$	$G_y = -S\alpha_j F_z(q_k)$	2	1
		$C\alpha_j \neq 0$	$(G_x - d_j)^2 + \left[\frac{G_y + S\alpha_j F_z}{C\alpha_j} \right]^2 = F_x^2 + F_y^2$	11	9
P	P		$F_x + d_j = G_x$	2	1

Table 4.4. Type of equations and maximum number of solutions for q_i , q_j and q_k

i	j	Conditions	Type / Number of solutions				Order
			θ_k	r_k	q_i	q_j	
R	R	$S\alpha_j = 0$	2 / 2	1 / 1	3 / 1	2 / 2	θ_j then θ_i
		$d_j = 0$	2 / 2	9 / 2	3 / 1	3 / 2	θ_j then θ_i
		$d_j \neq 0$ and $S\alpha_j \neq 0$	11 / 4	10 / 4	3 / 1	3 / 1	θ_i then θ_j
R	P	$C\alpha_j = 0$	2 / 2	1 / 1	2 / 2	1 / 1	θ_i then r_j
		$C\alpha_j \neq 0$	11 / 4	9 / 2	3 / 1	1 / 1	θ_i then r_j
P	R	$C\alpha_j = 0$	2 / 2	1 / 1	1 / 1	2 / 2	θ_j then r_i
		$C\alpha_j \neq 0$	11 / 4	9 / 2	1 / 1	3 / 1	θ_j then r_i
P	P		2 / 2	1 / 1	1 / 1	1 / 1	r_j then r_i

• **Example 4.3.** Solving q_k when joints i and j are revolute. In this case, equation [4.15] is written as:

$$\mathbf{Rot}(z, \theta_j) \mathbf{Rot}(x, \alpha_j) \mathbf{Trans}(x, d_j) \mathbf{Rot}(z, \theta_j) \begin{bmatrix} \mathbf{F}(q_k) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ 1 \end{bmatrix} \quad [4.17]$$

Postmultiplying equation [4.17] by $\mathbf{Rot}(z, -\theta_j)$, we obtain:

$$\begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & 0 \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \\ 1 \end{bmatrix} = \begin{bmatrix} C\theta_i & S\theta_i & 0 & 0 \\ -S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} G_x \\ G_y \\ G_z \\ 1 \end{bmatrix} \quad [4.18]$$

Expanding equation [4.18] gives:

$$C\theta_j F_x - S\theta_j F_y + d_j = C\theta_i G_x + S\theta_i G_y \quad [4.18a]$$

$$C\alpha_j S\theta_j F_x + C\alpha_j C\theta_j F_y - S\alpha_j F_z = -S\theta_i G_x + C\theta_i G_y \quad [4.18b]$$

$$S\alpha_j S\theta_j F_x + S\alpha_j C\theta_j F_y + C\alpha_j F_z = G_z \quad [4.18c]$$

Three cases are considered depending on the values of the geometric parameters α_j and d_j :

a) $S\alpha_j = 0$ (thus $C\alpha_j = \pm 1$), $d_j \neq 0$. Equation [4.18c] can be written as:

$$C\alpha_j F_z(q_k) = G_z \quad [4.19]$$

We thus deduce that:

- if $q_k = \theta_k$, equation [4.19] is of type 2 in θ_k ;
- if $q_k = r_k$, equation [4.19] is of type 1 in r_k .

Having determined q_k , the components of \mathbf{F} are considered to be known. Adding the squares of equations [4.18a] and [4.18b] eliminates θ_i and gives a type-2 equation in θ_j :

$$F_x^2 + F_y^2 + d_j^2 + 2 d_j (C\theta_j F_x - S\theta_j F_y) = G_x^2 + G_y^2 \quad [4.20]$$

After obtaining θ_j , equations [4.18a] and [4.18b] give a system of type-3 equations in θ_i .

b) $d_j = 0$ and $S\alpha_j \neq 0$. Adding the squares of equations [4.18] gives:

$$\|\mathbf{F}\|^2 = \|\mathbf{G}\|^2 \quad [4.21]$$

Note that $\|\mathbf{F}\|^2$ is a function of q_k whereas $\|\mathbf{G}\|^2$ is a constant:

- if $q_k = \theta_k$, equation [4.21] is of type 2 in θ_k ;
- if $q_k = r_k$, equation [4.21] is of type 9 in r_k .

Having obtained q_k and \mathbf{F} , equation [4.18c] gives θ_j using the type-2 equation. Finally, equations [4.18a] and [4.18b] give a system of type-3 equations in θ_i .

c) $d_j \neq 0$ and $S\alpha_j \neq 0$. Writing equation [4.17] in the form:

$$\begin{bmatrix} \mathbf{F}(q_k) \\ 1 \end{bmatrix} = \mathbf{Rot}(\mathbf{z}, -\theta_j) \mathbf{Trans}(\mathbf{x}, -d_j) \mathbf{Rot}(\mathbf{x}, -\alpha_j) \mathbf{Rot}(\mathbf{z}, -\theta_i) \begin{bmatrix} \mathbf{G} \\ 1 \end{bmatrix} \quad [4.22]$$

after expanding, we obtain the third component as:

$$F_z = S\alpha_j S\theta_i G_x - S\alpha_j C\theta_i G_y + C\alpha_j G_z \quad [4.23a]$$

Adding the squares of the components of equation [4.22] eliminates θ_j :

$$\|\mathbf{G}\|^2 + d_j^2 - 2 d_j (C\theta_i G_x - S\theta_i G_y) = \|\mathbf{F}\|^2 \quad [4.23b]$$

By eliminating θ_i from equations [4.23], we obtain:

$$\left[\frac{\|\mathbf{F}\|^2 - \|\mathbf{G}\|^2 - d_j^2}{2d_j} \right]^2 + \left[\frac{F_z - C\alpha_j G_z}{S\alpha_j} \right]^2 = G_x^2 + G_y^2 \quad [4.24]$$

Here, we distinguish two cases:

- if $q_k = \theta_k$, equation [4.24] is of type 11 in θ_k ;
- if $q_k = r_k$, equation [4.24] is of type 10 in r_k .

Knowing θ_k , equations [4.23a] and [4.23b] give a system of type-3 equations in θ_i . Finally, equations [4.18a] and [4.18b] are of type 3 in θ_j .

• **Example 4.4.** The variables θ_1 , θ_2 , θ_3 for the Stäubli RX-90 robot can be determined with the following equations using the Pieper method:

- equation for θ_3 : $-2D3 RL4 S3 = (P_x)^2 + (P_y)^2 + (P_z)^2 - (D3)^2 - (RL4)^2$
- equation for θ_2 : $(-RL4 S3 + D3) S2 + (RL4 C3) C2 = P_z$
- equations for θ_1 : $[(-RL4 S3 + D3) C2 - RL4 C3 S2] C1 = P_x$

$$[(-RL4\ S3 + D3)\ C2 - RL4\ C3\ S2]\ S1 = P_y$$

4.4.2.2. General solution of the orientation equation

The spherical joint variables θ_{m-1} , θ_m and θ_{m+1} are determined from the orientation equation, which is deduced from equation [4.2] as:

$${}^0\mathbf{R}_{m-2} {}^{m-2}\mathbf{R}_{m+1}(\theta_{m-1}, \theta_m, \theta_{m+1}) {}^{m+1}\mathbf{R}_6 = [\mathbf{s} \ \mathbf{n} \ \mathbf{a}] \quad [4.25]$$

The matrices ${}^0\mathbf{R}_{m-2}$ and ${}^{m+1}\mathbf{R}_6$ are functions of the variables that have already been obtained. Using equation [3.3] and after rearranging, equation [4.25] becomes:

$$\mathbf{rot}(\mathbf{z}, \theta_{m-1}) \mathbf{rot}(\mathbf{x}, \alpha_m) \mathbf{rot}(\mathbf{z}, \theta_m) \mathbf{rot}(\mathbf{x}, \alpha_{m+1}) \mathbf{rot}(\mathbf{z}, \theta_{m+1}) = [\mathbf{S} \ \mathbf{N} \ \mathbf{A}] \quad [4.26]$$

$$\text{with } [\mathbf{S} \ \mathbf{N} \ \mathbf{A}] = \mathbf{rot}(\mathbf{x}, -\alpha_{m-1}) {}^{m-2}\mathbf{R}_0 [\mathbf{s} \ \mathbf{n} \ \mathbf{a}] {}^6\mathbf{R}_{m+1}$$

The left side of equation [4.26] is a function of the joint variables θ_{m-1} , θ_m and θ_{m+1} whereas the right side is known. Since $\mathbf{rot}(\mathbf{z}, \theta)$ defines a rotation about the axis $\mathbf{z}_0 = [0 \ 0 \ 1]^T$, then \mathbf{z}_0 is invariant with this rotation, which results in:

$$\mathbf{rot}(\mathbf{z}, \theta) \mathbf{z}_0 = \mathbf{z}_0 \text{ and } \mathbf{z}_0^T \mathbf{rot}(\mathbf{z}, \theta) = \mathbf{z}_0^T \quad [4.27]$$

i) determination of θ_m

To eliminate θ_{m-1} , we premultiply equation [4.26] by \mathbf{z}_0^T and postmultiply it by \mathbf{z}_0 :

$$\mathbf{z}_0^T \mathbf{rot}(\mathbf{x}, \alpha_m) \mathbf{rot}(\mathbf{z}, \theta_m) \mathbf{rot}(\mathbf{x}, \alpha_{m+1}) \mathbf{z}_0 = \mathbf{z}_0^T [\mathbf{S} \ \mathbf{N} \ \mathbf{A}] \mathbf{z}_0 \quad [4.28]$$

thus, we obtain:

$$S\alpha_m S\alpha_{m+1} C\theta_m + C\alpha_m C\alpha_{m+1} = A_z$$

Equation [4.28] is of type 2 in θ_m and gives two solutions (Appendix 1);

ii) determination of θ_{m-1}

Having obtained θ_m , let us write:

$$[\mathbf{S1} \ \mathbf{N1} \ \mathbf{A1}] = \mathbf{rot}(\mathbf{x}, \alpha_m) \mathbf{rot}(\mathbf{z}, \theta_m) \mathbf{rot}(\mathbf{x}, \alpha_{m+1}) \quad [4.29]$$

Postmultiplying equation [4.26] by \mathbf{z}_0 and using equation [4.29] gives:

$$\mathbf{rot}(\mathbf{z}, \theta_{m-1}) \mathbf{A1} = \mathbf{A} \quad [4.30]$$

The first two elements of [4.30] give a type-3 system of equations in θ_{m-1} ;

iii) determination of θ_{m+1}

By premultiplying equation [4.26] by \mathbf{z}_0^T and using equation [4.29], we obtain:

$$\begin{bmatrix} S1_z & N1_z & A1_z \end{bmatrix} \mathbf{rot}(\mathbf{z}, \theta_{m+1}) = \begin{bmatrix} S_z & N_z & A_z \end{bmatrix} \quad [4.31]$$

This gives a type-3 system of equations in θ_{m+1} .

These equations yield two solutions for the spherical joint variables. Thus, the maximum number of solutions of the IGM for a six degree-of-freedom robot with a spherical joint is eight.

4.4.3. Inverse geometric model of robots with three prismatic joints

The IGM of this class of robots is obtained by solving firstly the three revolute joint variables using the orientation equation. After this, the prismatic joint variables are obtained using the position equation. The number of solutions for the IGM of such robots is two.

4.4.3.1. Solution of the orientation equation

Let the revolute joints be denoted by i, j and k. The orientation equation can be deduced from equations [4.2] and [3.3] as:

$$\mathbf{rot}(\mathbf{z}, \theta_i) \begin{bmatrix} S1 & N1 & A1 \end{bmatrix} \mathbf{rot}(\mathbf{z}, \theta_j) \begin{bmatrix} S2 & N2 & A2 \end{bmatrix} \mathbf{rot}(\mathbf{z}, \theta_k) = \begin{bmatrix} S3 & N3 & A3 \end{bmatrix} \quad [4.32]$$

where the orientation matrices $\begin{bmatrix} S_i & N_i & A_i \end{bmatrix}$, for $i = 1, 2, 3$, are known. The solution of equation [4.32] is similar to that of § 4.4.2.2 and gives two solutions.

4.4.3.2. Solution of the position equation

Let the prismatic joints be denoted by i' , j' and k' . The revolute joint variables being determined, the position equation is written as:

$$\text{Trans}(z, r_{i'})T_1 \text{Trans}(z, r_{j'})T_2 \text{Trans}(z, r_{k'})=T_3 \quad [4.33]$$

$$\text{with } T_i = \begin{bmatrix} S_i & N_i & A_i & P_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrices T_i , for $i = 1, 2, 3$, are known. The fourth column elements of the previous equation gives a system of three linear equations in $r_{i'}$, $r_{j'}$ and $r_{k'}$.

4.5. Inverse geometric model of general 6 dof robots

The Raghavan-Roth method [Raghavan 90] gives the solution to the inverse geometric problem for six degree-of-freedom robots with general geometry (the geometric parameters may have arbitrary real values). In this method, we first compute all possible solutions of one variable q_i using a polynomial equation, which is called the *characteristic polynomial*. Then, the other variables are uniquely derived for each q_i . This method is based on the *dalytic elimination* technique presented in Appendix 3.

In order to illustrate this method, we consider the 6R robot and rewrite equation [4.2] as follows:

$${}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 = U_0 {}^6T_5 {}^5T_4 \quad [4.34]$$

The left and right sides of equation [4.34] represent the transformation of frame R_4 relative to frame R_0 using two distinct paths. The joint variables appearing in the elements of the previous equation are:

$$\begin{bmatrix} \theta_1, \theta_2, \theta_3, \theta_4 & \theta_1, \theta_2, \theta_3, \theta_4 & \theta_1, \theta_2, \theta_3 & \theta_1, \theta_2, \theta_3 \\ \theta_1, \theta_2, \theta_3, \theta_4 & \theta_1, \theta_2, \theta_3, \theta_4 & \theta_1, \theta_2, \theta_3 & \theta_1, \theta_2, \theta_3 \\ \theta_1, \theta_2, \theta_3, \theta_4 & \theta_1, \theta_2, \theta_3, \theta_4 & \theta_1, \theta_2, \theta_3 & \theta_1, \theta_2, \theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \theta_5, \theta_6 & \theta_5, \theta_6 & \theta_5, \theta_6 & \theta_5, \theta_6 \\ \theta_5, \theta_6 & \theta_5, \theta_6 & \theta_5, \theta_6 & \theta_5, \theta_6 \\ \theta_5, \theta_6 & \theta_5, \theta_6 & \theta_5, \theta_6 & \theta_5, \theta_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From this equation, we observe that the third and fourth columns of the left side are independent of θ_4 . This is due to the fact that the elements of the third and fourth columns of the transformation matrix ${}^{i-1}\mathbf{T}_i$ are independent of θ_i (see equation [3.2]). From equation [4.34], we can thus establish the following equations devoid of θ_4 :

$$\mathbf{a}_l = \mathbf{a}_r \quad [4.35a]$$

$$\mathbf{P}_l = \mathbf{P}_r \quad [4.35b]$$

where the vectors \mathbf{a} and \mathbf{P} contain the first three elements of the third and fourth columns of equation [4.34] respectively, and the subscripts "l" and "r" indicate the left and right sides respectively. Equations [4.35] give six scalar equations.

It is now necessary to eliminate four of the five remaining variables to obtain a polynomial equation in one variable. This requires the use of the following additional equations:

$$(\mathbf{a}^T \mathbf{P})_l = (\mathbf{a}^T \mathbf{P})_r \quad [4.36a]$$

$$(\mathbf{P}^T \mathbf{P})_l = (\mathbf{P}^T \mathbf{P})_r \quad [4.36b]$$

$$(\mathbf{a} \times \mathbf{P})_l = (\mathbf{a} \times \mathbf{P})_r \quad [4.36c]$$

$$[\mathbf{a}(\mathbf{P}^T \mathbf{P}) - 2\mathbf{P}(\mathbf{a}^T \mathbf{P})]_l = [\mathbf{a}(\mathbf{P}^T \mathbf{P}) - 2\mathbf{P}(\mathbf{a}^T \mathbf{P})]_r \quad [4.36d]$$

Equations [4.36a] and [4.36b] are scalar, whereas equations [4.36c] and [4.36d] are vectors. They do not contain $\sin^2(\cdot)$, $\cos^2(\cdot)$ or $\sin(\cdot)\cos(\cdot)$. We thus have fourteen scalar equations that may be written in the following matrix form:

$$\mathbf{A} \mathbf{X1} = \mathbf{B} \mathbf{Y} \quad [4.37]$$

where:

$$\bullet \mathbf{X1} = [S2S3 \ S2C3 \ C2S3 \ C2C3 \ S2 \ C2 \ S3 \ C3 \ 1]^T \quad [4.38]$$

$$\bullet \mathbf{Y} = [S5S6 \ S5C6 \ C5S6 \ C5C6 \ S5 \ C5 \ S6 \ C6]^T \quad [4.39]$$

• \mathbf{A} : (14x9) matrix whose elements are linear combinations of $S1$ and $C1$;

• \mathbf{B} : (14x8) matrix whose elements are constants.

To eliminate θ_5 and θ_6 , we select eight scalar equations out of equation [4.37]. The system [4.37] will be partitioned as:

$$\begin{bmatrix} \mathbf{A1} \\ \mathbf{A2} \end{bmatrix} \mathbf{X1} = \begin{bmatrix} \mathbf{B1} \\ \mathbf{B2} \end{bmatrix} \mathbf{Y} \quad [4.40]$$

where $\mathbf{A1} \mathbf{X1} = \mathbf{B1} \mathbf{Y}$ gives six equations, and $\mathbf{A2} \mathbf{X1} = \mathbf{B2} \mathbf{Y}$ represents the remaining eight equations. By eliminating \mathbf{Y} , we obtain the following system of equations:

$$\mathbf{D} \mathbf{X1} = \mathbf{0}_{6 \times 1} \quad [4.41]$$

where $\mathbf{D} = [\mathbf{A1} - \mathbf{B1} \mathbf{B2}^{-1} \mathbf{A2}]$ is a (6x9) matrix whose elements are functions of $S1$ and $C1$.

Using the half-angle transformation for the sine and cosine functions in equation [4.41] ($Ci = \frac{1 - x_i^2}{1 + x_i^2}$ and $Si = \frac{2x_i}{1 + x_i^2}$ with $x_i = \tan \frac{\theta_i}{2}$ for $i = 1, 2, 3$) yields the new homogeneous system of equations:

$$\mathbf{E} \mathbf{X2} = \mathbf{0}_{6 \times 1} \quad [4.42]$$

where \mathbf{E} is a (6x9) matrix whose elements are quadratic functions of x_1 , and:

$$\mathbf{X2} = [x_2^2 x_3^2 \quad x_2^2 x_3 \quad x_2^2 \quad x_2 x_3^2 \quad x_2 x_3 \quad x_2 \quad x_3^2 \quad x_3 \quad 1]^T \quad [4.43]$$

Thus, we have a system of six equations with nine unknowns. We now eliminate x_2 and x_3 dyalitically (see Appendix 3). Multiplying equation [4.42] by x_2 , we obtain six additional equations with only three new unknowns:

$$\mathbf{E} \mathbf{X3} = \mathbf{0}_{6 \times 1} \quad [4.44]$$

with $\mathbf{X3} = [x_2^3 x_3^2 \quad x_2^3 x_3 \quad x_2^3 \quad x_2^2 x_3^2 \quad x_2^2 x_3 \quad x_2^2 \quad x_2 x_3^2 \quad x_2 x_3 \quad x_2]^T$.

Combining equations [4.42] and [4.44], we obtain a system of twelve homogeneous equations:

$$\mathbf{S} \mathbf{X} = \mathbf{0}_{12 \times 1} \quad [4.45]$$

where:

$$\mathbf{X} = [x_2^3 x_3^2 \quad x_2^3 x_3 \quad x_2^3 \quad x_2^2 x_3^2 \quad x_2^2 x_3 \quad x_2^2 \quad x_2 x_3^2 \quad x_2 x_3 \quad x_2 \quad x_3^2 \quad x_3 \quad 1]^T \quad [4.46]$$

and \mathbf{S} is a (12x12) matrix whose elements are quadratic functions of x_1 and has the following form:

$$\mathbf{S} = \begin{bmatrix} \mathbf{E} & \mathbf{0}_{6 \times 3} \\ \mathbf{0}_{6 \times 3} & \mathbf{E} \end{bmatrix} \quad [4.47]$$

In order that equation [4.45] has a non-trivial solution, the determinant of the matrix \mathbf{S} must be zero. The characteristic polynomial of equation [4.47], which gives the solution for x_1 , can be obtained from:

$$\det(\mathbf{S}) = 0 \quad [4.48]$$

It can be shown that this determinant, which is a polynomial of degree 24, has $(1+x_1^2)^4$ as a common factor [Raghavan 90]. Thus, equation [4.48] is written as:

$$\det(\mathbf{S}) = f(x_1) (1+x_1^2)^4 = 0 \quad [4.49]$$

The polynomial $f(x_1)$ is of degree sixteen and represents the characteristic polynomial of the robot. The real roots of this polynomial give all the solutions for θ_1 . For each value of θ_1 , we can calculate the matrix \mathbf{S} . The variables θ_2 and θ_3 are uniquely determined by solving the linear system of equation [4.45]. By substituting θ_1 , θ_2 and θ_3 in equation [4.37] and using eight equations, we can calculate θ_5 and θ_6 . Finally, we consider the following equation to calculate θ_4 :

$${}^4\mathbf{T}_3 = {}^4\mathbf{T}_6 \mathbf{U}_0 {}^0\mathbf{T}_3 \quad [4.50]$$

By using the (1, 1) and (2, 1) elements, we obtain θ_4 using an atan2 function.

The same method can also be applied to six degree-of-freedom robots having prismatic joints. In this case, S_i and C_i have to be replaced by r_i^2 and r_i in $\mathbf{X1}$ and \mathbf{Y} respectively, i being the prismatic joint.

NOTE.— Equation [4.34] is a particular form of equation [4.2] that can be written in several other forms [Mavroidis 93], for example:

$${}^4\mathbf{T}_5 {}^5\mathbf{T}_6 {}^6\mathbf{T}_7 {}^0\mathbf{T}_1 = {}^4\mathbf{T}_3 {}^3\mathbf{T}_2 {}^2\mathbf{T}_1 \quad [4.51a]$$

$${}^5\mathbf{T}_6 {}^6\mathbf{T}_7 {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 = {}^5\mathbf{T}_4 {}^4\mathbf{T}_3 {}^3\mathbf{T}_2 \quad [4.51b]$$

$${}^6\mathbf{T}_7 {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 = {}^6\mathbf{T}_5 {}^5\mathbf{T}_4 {}^4\mathbf{T}_3 \quad [4.51c]$$

$${}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 {}^3\mathbf{T}_4 = {}^7\mathbf{T}_6 {}^6\mathbf{T}_5 {}^5\mathbf{T}_4 \quad [4.51d]$$

$${}^1\mathbf{T}_2 {}^2\mathbf{T}_3 {}^3\mathbf{T}_4 {}^4\mathbf{T}_5 = {}^1\mathbf{T}_0 {}^7\mathbf{T}_6 {}^6\mathbf{T}_5 \quad [4.51e]$$

$${}^2\mathbf{T}_3 {}^3\mathbf{T}_4 {}^4\mathbf{T}_5 {}^5\mathbf{T}_6 = {}^2\mathbf{T}_1 {}^1\mathbf{T}_0 {}^7\mathbf{T}_6 \quad [4.51f]$$

with ${}^7\mathbf{T}_6 = \mathbf{U}_0$ and ${}^6\mathbf{T}_7 = \mathbf{U}_0^{-1}$

The selection of the starting equation not only defines the variable of the characteristic equation but also the degree of the corresponding polynomial. For specific values of the geometric parameters, certain columns of the matrix \mathbf{S} become dependent and it is necessary to either change the selected variables and columns [Khalil 94b], [Murareci 97] or choose another starting equation [Mavroidis 93].

When the robot is in a singular configuration, the rows of the matrix \mathbf{S} are linearly dependent. In this case, it is not possible to find a solution. In fact this method has proved the maximum number of solutions that can be obtained for the inverse geometric problem of serial robots, but it is difficult to use it to develop a general numerical model to treat any robot.

4.6. Conclusion

In this chapter, we have presented three methods for calculating the inverse geometric model. The Paul method is applicable to a large number of structures with particular geometrical parameters where most of the distances are zero and most of the angles are zero or $\pm \pi/2$. The Pieper method gives the solution for the six degree-of-freedom robots having three prismatic joints or three revolute joints whose axes intersect at a point. Finally, the general method provides the solution for the IGM of six degree-of-freedom robots with general geometry.

The analytical solution, as compared to the differential methods discussed in the next chapter, is useful for obtaining all the solutions of the inverse geometric model. Some of them may be eliminated because they do not satisfy the joint limits. Generally, the selected solution is left to the robot's user and depends on the task specifications: to avoid collisions between the robot and its environment; to ensure the continuity of the trajectory as required in certain tasks prohibiting configuration changes (machining, welding,...); to avoid as much as possible the singular configurations that may induce control problems (namely discontinuity of velocity), etc.

Chapter 5

Direct kinematic model of serial robots

5.1. Introduction

The direct kinematic model of a robot manipulator gives the velocity of the end-effector $\dot{\mathbf{X}}$ in terms of the joint velocities $\dot{\mathbf{q}}$. It is written as:

$$\dot{\mathbf{X}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad [5.1]$$

where $\mathbf{J}(\mathbf{q})$ denotes the (mxn) Jacobian matrix.

The same Jacobian matrix also appears in the direct differential model, which provides the differential displacement of the end-effector $d\mathbf{X}$ in terms of the differential variation of the joint variables $d\mathbf{q}$:

$$d\mathbf{X} = \mathbf{J}(\mathbf{q}) d\mathbf{q} \quad [5.2]$$

The Jacobian matrix has multiple applications in robotics [Whitney 69], [Paul 81]. The most obvious is the use of its inverse to numerically compute a solution for the inverse geometric model, i.e. to compute the joint variables \mathbf{q} corresponding to a given location of the end-effector \mathbf{X} (Chapter 6). The transpose of the Jacobian matrix is used in the static model to compute the necessary joint forces and torques to exert specified forces and moments on the environment. The Jacobian matrix is also used to determine the degrees of freedom of the tool, the singularities and to analyze the reachable workspace of robots [Borrel 86], [Wenger 89].

In this chapter, we will present the computation of the Jacobian matrix and expose its different applications for serial robots. The kinematic model of complex chain robots will be studied in Chapter 7.

5.2. Computation of the Jacobian matrix from the direct geometric model

The Jacobian matrix can be obtained by differentiating the DGM, $\mathbf{X} = \mathbf{f}(\mathbf{q})$, using the partial derivative $\frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ such that:

$$J_{ij} = \frac{\partial f_i(\mathbf{q})}{\partial q_j}; \quad \text{for } i = 1, \dots, m \text{ and } j = 1, \dots, n \quad [5.3]$$

where J_{ij} is the (i, j) element of the Jacobian matrix \mathbf{J} .

This method is convenient for simple robots having a reduced number of degrees of freedom as shown in the following example. The computation of the kinematic Jacobian matrix, is more practical for a general n degree-of-freedom robot. It is presented in § 5.3.

• **Example 5.1.** Let us consider the three degree-of-freedom planar robot presented in Figure 5.1. Let us denote the link lengths by $L1$, $L2$ and $L3$.

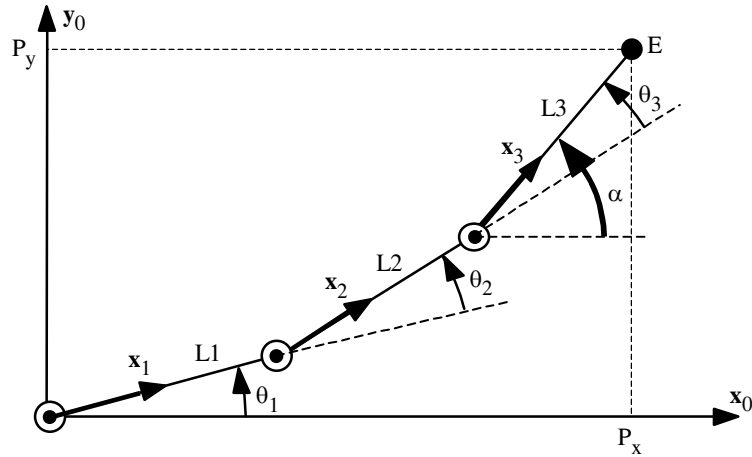


Figure 5.1. Example of a three degree-of-freedom planar robot

The task coordinates, defined as the position coordinates (P_x, P_y) of the terminal point E and the angle α giving the orientation of the third link relative to frame R_0 , are such that:

$$\begin{aligned} P_x &= C1 L1 + C12 L2 + C123 L3 \\ P_y &= S1 L1 + S12 L2 + S123 L3 \end{aligned}$$

$$\alpha = \theta_1 + \theta_2 + \theta_3$$

where $C1 = \cos(\theta_1)$, $S1 = \sin(\theta_1)$, $C12 = \cos(\theta_1 + \theta_2)$, $S12 = \sin(\theta_1 + \theta_2)$, $C123 = \cos(\theta_1 + \theta_2 + \theta_3)$ and $S123 = \sin(\theta_1 + \theta_2 + \theta_3)$.

The Jacobian matrix is obtained by differentiating these expressions with respect to θ_1 , θ_2 and θ_3 :

$$\mathbf{J} = \begin{bmatrix} -S1L1 - S12L2 - S123L3 & -S12L2 - S123L3 & -S123L3 \\ C1L1 + C12L2 + C123L3 & C12L2 + C123L3 & C123L3 \\ 1 & 1 & 1 \end{bmatrix}$$

By taking as operational coordinates the position coordinates of point O_3 and the angle α , the Jacobian matrix will be simplified as:

$$\mathbf{J} = \begin{bmatrix} -S1L1 - S12L2 & -S12L2 & 0 \\ C1L1 + C12L2 & C12L2 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

5.3. Kinematic Jacobian matrix

In this section, we present a direct method to compute the Jacobian matrix of a serial mechanism without differentiating the DGM. The Jacobian matrix obtained is called the *kinematic Jacobian matrix*. It gives the kinematic screw of frame R_n in terms of the joint velocities $\dot{\mathbf{q}}$:

$$\mathbf{V}_n = \mathbf{J}_n \dot{\mathbf{q}} \quad [5.4a]$$

$$\mathbf{V}_n = \begin{bmatrix} \mathbf{v}_n \\ \boldsymbol{\omega}_n \end{bmatrix}$$

where \mathbf{v}_n and $\boldsymbol{\omega}_n$ are the linear and angular velocities of frame R_n respectively. We note that \mathbf{v}_n is the derivative of the position vector \mathbf{P}_n with respect to time, while $\boldsymbol{\omega}_n$ is not the derivative of any orientation vector.

The kinematic Jacobian matrix also gives the relationship between the differential translation and rotation vectors ($d\mathbf{P}_n$, $\boldsymbol{\delta}_n$) of frame R_n in terms of the differential joint variables $d\mathbf{q}$:

$$\begin{bmatrix} \mathbf{dP}_n \\ \delta_n \end{bmatrix} = \mathbf{J}_n \mathbf{dq} \quad [5.4b]$$

We will show in § 5.11 that the Jacobian giving the end-effector velocity $\dot{\mathbf{X}}$, for any task coordinate representation, can be deduced from the kinematic Jacobian \mathbf{J}_n .

5.3.1. Computation of the kinematic Jacobian matrix

The velocity \dot{q}_k of joint k produces the linear and angular velocities ($\mathbf{v}_{k,n}$ and $\omega_{k,n}$ respectively) at the terminal frame R_n . Two cases are considered:

- if joint k is prismatic ($\sigma_k = 1$, Figure 5.2):

$$\begin{bmatrix} \mathbf{v}_{k,n} \\ \omega_{k,n} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_k \dot{q}_k \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_k \\ \mathbf{0} \end{bmatrix} \dot{q}_k \quad [5.5]$$

where \mathbf{a}_k is the unit vector along the z_k axis;

- if joint k is revolute ($\sigma_k = 0$, Figure 5.3):

$$\begin{bmatrix} \mathbf{v}_{k,n} \\ \omega_{k,n} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_k \dot{q}_k \times \mathbf{L}_{k,n} \\ \mathbf{a}_k \dot{q}_k \end{bmatrix} = \begin{bmatrix} \mathbf{a}_k \times \mathbf{L}_{k,n} \\ \mathbf{a}_k \end{bmatrix} \dot{q}_k \quad [5.6]$$

where $\mathbf{L}_{k,n}$ denotes the position vector connecting O_k to O_n .

Thus, $\mathbf{v}_{k,n}$ and $\omega_{k,n}$ can be written in the following general form:

$$\begin{bmatrix} \mathbf{v}_{k,n} \\ \omega_{k,n} \end{bmatrix} = \begin{bmatrix} \sigma_k \mathbf{a}_k + \bar{\sigma}_k (\mathbf{a}_k \times \mathbf{L}_{k,n}) \\ \bar{\sigma}_k \mathbf{a}_k \end{bmatrix} \dot{q}_k \quad [5.7]$$

The linear and angular velocities of the terminal frame can be written as:

$$\begin{bmatrix} \mathbf{v}_n \\ \omega_n \end{bmatrix} = \sum_{k=1}^n \begin{bmatrix} \sigma_k \mathbf{a}_k + \bar{\sigma}_k (\mathbf{a}_k \times \mathbf{L}_{k,n}) \\ \bar{\sigma}_k \mathbf{a}_k \end{bmatrix} \dot{q}_k \quad [5.8]$$

Writing equation [5.8] in matrix form and using equation [5.4], we deduce that:

$$\mathbf{J}_n = \begin{bmatrix} \sigma_1 \mathbf{a}_1 + \bar{\sigma}_1 (\mathbf{a}_1 \times \mathbf{L}_{1,n}) & \dots & \sigma_n \mathbf{a}_n + \bar{\sigma}_n (\mathbf{a}_n \times \mathbf{L}_{n,n}) \\ \bar{\sigma}_1 \mathbf{a}_1 & \dots & \bar{\sigma}_n \mathbf{a}_n \end{bmatrix} \quad [5.9]$$

Expressing the vectors of \mathbf{J}_n with respect to frame R_i , we obtain the (6xn) Jacobian matrix ${}^i\mathbf{J}_n$ such that:

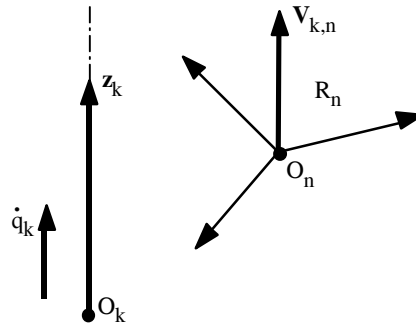


Figure 5.2. Case of a prismatic joint

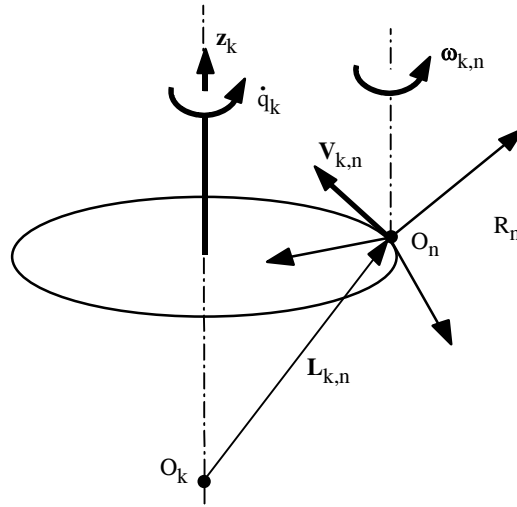


Figure 5.3. Case of a revolute joint

$${}^i\mathbf{V}_n = {}^i\mathbf{J}_n \dot{\mathbf{q}} \quad [5.10]$$

In general, we calculate \mathbf{v}_n and $\boldsymbol{\omega}_n$ in frame \mathbf{R}_n or frame \mathbf{R}_0 . The corresponding Jacobian matrix is denoted by ${}^n\mathbf{J}_n$ or ${}^0\mathbf{J}_n$ respectively. These matrices can also be computed using any matrix ${}^i\mathbf{J}_n$, for $i = 0, \dots, n$, thanks to the following expression:

$${}^s\mathbf{J}_n = \begin{bmatrix} {}^s\mathbf{R}_i & \mathbf{0}_3 \\ \mathbf{0}_3 & {}^s\mathbf{R}_i \end{bmatrix} {}^i\mathbf{J}_n \quad [5.11]$$

where ${}^s\mathbf{R}_i$ is the (3x3) orientation matrix of frame \mathbf{R}_i relative to frame \mathbf{R}_s .

In general, we obtain the simplest matrix ${}^i\mathbf{J}_n$ when $i = \text{integer}(n/2)$. We note that the matrices ${}^i\mathbf{J}_n$, for $i = 0, \dots, n$, have the same singular positions.

5.3.2. Computation of the matrix ${}^i\mathbf{J}_n$

Since the vector product $\mathbf{a}_k \times \mathbf{L}_{k,n}$ can be computed by $\hat{\mathbf{a}}_k \mathbf{L}_{k,n}$, the k^{th} column of ${}^i\mathbf{J}_n$, denoted as ${}^i\mathbf{J}_n(:,k)$, becomes:

$${}^i\mathbf{J}_n(:,k) = \begin{bmatrix} \sigma_k^i \mathbf{a}_k + \bar{\sigma}_k^i \mathbf{R}_k^k \hat{\mathbf{a}}_k \mathbf{L}_{k,n} \\ \bar{\sigma}_k^i \mathbf{a}_k \end{bmatrix}$$

Since ${}^k\mathbf{a}_k = [0 \quad 0 \quad 1]^T$ and ${}^k\mathbf{L}_{k,n} = {}^k\mathbf{P}_n$, we obtain:

$${}^i\mathbf{J}_n(:,k) = \begin{bmatrix} \sigma_k^i \mathbf{a}_k + \bar{\sigma}_k^i (-{}^kP_{n_y} {}^i\mathbf{s}_k + {}^kP_{n_x} {}^i\mathbf{n}_k) \\ \bar{\sigma}_k^i \mathbf{a}_k \end{bmatrix} \quad [5.12]$$

where ${}^kP_{n_x}$ and ${}^kP_{n_y}$ denote the x and y components of the vector ${}^k\mathbf{P}_n$ respectively.

From this expression, we obtain the k^{th} column of ${}^n\mathbf{J}_n$ as:

$${}^n\mathbf{J}_n(:,k) = \begin{bmatrix} \sigma_k^n \mathbf{a}_k + \bar{\sigma}_k^n (-{}^kP_{n_y} {}^n\mathbf{s}_k + {}^kP_{n_x} {}^n\mathbf{n}_k) \\ \bar{\sigma}_k^n \mathbf{a}_k \end{bmatrix} \quad [5.13]$$

The column ${}^n\mathbf{J}_n(:,k)$ is computed from the elements of the matrix ${}^k\mathbf{T}_n$ resulting from the DGM.

In a similar way since $\mathbf{L}_{k,n} = \mathbf{L}_{k,i} + \mathbf{L}_{i,n} = \mathbf{L}_{i,n} - \mathbf{L}_{i,k}$, thus the k^{th} column of ${}^i\mathbf{J}_n$ can be written as:

$${}^i\mathbf{J}_n(:,k) = \begin{bmatrix} \sigma_k^i \mathbf{a}_k + \bar{\sigma}_k^i \hat{\mathbf{a}}_k ({}^i\mathbf{P}_n - {}^i\mathbf{P}_k) \\ \bar{\sigma}_k^i \mathbf{a}_k \end{bmatrix} \quad [5.14]$$

which gives for $i = 0$:

$${}^0\mathbf{J}_n(:,k) = \begin{bmatrix} \sigma_k^0 \mathbf{a}_k + \bar{\sigma}_k^0 \hat{\mathbf{a}}_k ({}^0\mathbf{P}_n - {}^0\mathbf{P}_k) \\ \bar{\sigma}_k^0 \mathbf{a}_k \end{bmatrix} \quad [5.15]$$

In this case, we need to compute the matrices ${}^0\mathbf{T}_k$ for $k = 1, \dots, n$.

NOTE. – To find the Jacobian ${}^E\mathbf{J}_E$ defining the velocity of the tool frame \mathbf{R}_E , we can either make use of equation [5.9] after replacing $\mathbf{L}_{k,n}$ by $\mathbf{L}_{k,E}$, or compute ${}^E\mathbf{V}_E$ as a function of ${}^n\mathbf{V}_n$, and deduce ${}^E\mathbf{J}_E$. From § 2.4.3, we can see that:

$${}^E\mathbf{V}_E = {}^E\mathbf{S}_n {}^n\mathbf{V}_n = {}^E\mathbf{S}_n {}^n\mathbf{J}_n \dot{\mathbf{q}}$$

where ${}^E\mathbf{S}_n$ is the (6x6) screw transformation matrix defined in equation [2.47]. Consequently, we deduce that:

$${}^E\mathbf{J}_E = {}^E\mathbf{S}_n {}^n\mathbf{J}_n \quad [5.16]$$

with

$${}^E\mathbf{S}_n = \begin{bmatrix} {}^E\mathbf{R}_n & -{}^E\mathbf{R}_n {}^n\hat{\mathbf{P}}_E \\ \mathbf{0}_3 & {}^E\mathbf{R}_n \end{bmatrix} = \begin{bmatrix} {}^E\mathbf{R}_n & {}^E\hat{\mathbf{P}}_n {}^E\mathbf{R}_n \\ \mathbf{0}_3 & {}^E\mathbf{R}_n \end{bmatrix}$$

• **Example 5.2.** Compute the Jacobian matrix ${}^6\mathbf{J}_6$ of the Stäubli RX-90 robot. Using equation [5.13] and the matrices ${}^k\mathbf{T}_6$ resulting from the DGM, we obtain:

$$\begin{aligned} J(1,1) &= (-C6C5S4 - S6C4)(S23RL4 - C2D3) \\ J(2,1) &= (S6C5S4 - C6C4)(S23RL4 - C2D3) \\ J(3,1) &= S5S4(S23RL4 - C2D3) \\ J(4,1) &= (C6C5C4 - S6S4)S23 + C6S5C23 \\ J(5,1) &= (-S6C5C4 - C6S4)S23 - S6S5C23 \\ J(6,1) &= -S5C4S23 + C5C23 \\ J(1,2) &= (-C6C5C4 + S6S4)(RL4 - S3D3) + C6S5C3D3 \\ J(2,2) &= (S6C5C4 + C6S4)(RL4 - S3D3) - S6S5C3D3 \\ J(3,2) &= S5C4(RL4 - S3D3) + C5C3D3 \\ J(4,2) &= -C6C5S4 - S6C4 \\ J(5,2) &= S6C5S4 - C6C4 \\ J(6,2) &= S5S4 \\ J(1,3) &= (-C6C5C4 + S6S4)RL4 \\ J(2,3) &= (S6C5C4 + C6S4)RL4 \\ J(3,3) &= S5C4RL4 \\ J(4,3) &= -C6C5S4 - S6C4 \end{aligned}$$

$$\begin{aligned}
J(5,3) &= S6C5S4 - C6C4 \\
J(6,3) &= S5S4 \\
J(1,4) &= 0 \\
J(2,4) &= 0 \\
J(3,4) &= 0 \\
J(4,4) &= C6S5 \\
J(5,4) &= -S6S5 \\
J(6,4) &= C5 \\
J(1,5) &= 0 \\
J(2,5) &= 0 \\
J(3,5) &= 0 \\
J(4,5) &= -S6 \\
J(5,5) &= -C6 \\
J(6,5) &= 0 \\
J(1,6) &= 0 \\
J(2,6) &= 0 \\
J(3,6) &= 0 \\
J(4,6) &= 0 \\
J(5,6) &= 0 \\
J(6,6) &= 1
\end{aligned}$$

• **Example 5.3.** Determine the Jacobian matrix ${}^3\mathbf{J}_6$ of the Stäubli RX-90 robot. The column k of the matrix ${}^3\mathbf{J}_6$ for a revolute joint is obtained from equation [5.12] as:

$${}^3\mathbf{J}_6(:,k) = \begin{bmatrix} -{}^kP_{6y} {}^3\mathbf{s}_k + {}^kP_{6x} {}^3\mathbf{n}_k \\ {}^3\mathbf{a}_k \end{bmatrix}$$

The elements ${}^kP_{6y}$ and ${}^kP_{6x}$ are obtained from the DGM. The vectors ${}^3\mathbf{s}_k$, ${}^3\mathbf{n}_k$ and ${}^3\mathbf{a}_k$, for $k = 2, 3, 4$ and 6 , are deduced from the matrices ${}^3\mathbf{R}_2$, ${}^3\mathbf{R}_3$, ${}^3\mathbf{R}_4$ and ${}^3\mathbf{R}_6$, which are also computed for the DGM. The additional matrices to be computed are ${}^3\mathbf{R}_1$ and ${}^3\mathbf{R}_5$. Finally, we obtain:

$${}^3\mathbf{J}_6 = \begin{bmatrix} 0 & -RL4 + S3D3 & -RL4 & 0 & 0 & 0 \\ 0 & C3D3 & 0 & 0 & 0 & 0 \\ S23RL4 - C2D3 & 0 & 0 & 0 & 0 & 0 \\ S23 & 0 & 0 & 0 & S4 & -S5C4 \\ C23 & 0 & 0 & 1 & 0 & C5 \\ 0 & 1 & 1 & 0 & C4 & S5S4 \end{bmatrix}$$

5.4. Decomposition of the Jacobian matrix into three matrices

We have shown in equation [5.11] that the matrix ${}^s\mathbf{J}_n$ could be decomposed into two matrices; the first is always of full-rank and the second contains simple elements. Renaud [Renaud 80b] has shown that the Jacobian matrix could also be decomposed into three matrices: the first two are always of full-rank and their inverse is straightforward; the third is of the same rank as ${}^s\mathbf{J}_n$, but contains simpler elements.

Figure 5.4 illustrates the principle of the proposed method: the influence of the joint velocities is not calculated at the level of the terminal frame \mathbf{R}_n but at the level of an intermediate frame \mathbf{R}_j . Therefore, we define the Jacobian matrix $\mathbf{J}_{n,j}$ as:

$$\mathbf{J}_{n,j} = \begin{bmatrix} \sigma_1 \mathbf{a}_1 + \bar{\sigma}_1 (\mathbf{a}_1 \times \mathbf{L}_{1,j}) & \dots & \sigma_n \mathbf{a}_n + \bar{\sigma}_n (\mathbf{a}_n \times \mathbf{L}_{n,j}) \\ \bar{\sigma}_1 \mathbf{a}_1 & \dots & \bar{\sigma}_n \mathbf{a}_n \end{bmatrix} \quad [5.17]$$

The matrix $\mathbf{J}_{n,j}$ is equivalent to the Jacobian matrix defining the velocity of a frame fixed to link n and aligned instantaneously with frame \mathbf{R}_j . We can compute \mathbf{J}_n from $\mathbf{J}_{n,j}$ using the expression:

$$\mathbf{J}_n = \begin{bmatrix} \mathbf{I}_3 & -\hat{\mathbf{L}}_{j,n} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{J}_{n,j} \quad [5.18]$$

By projecting the elements of this equation into frame \mathbf{R}_i , we obtain:

$${}^i\mathbf{J}_n = \begin{bmatrix} \mathbf{I}_3 & -{}^i\hat{\mathbf{L}}_{j,n} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} {}^i\mathbf{J}_{n,j} \quad [5.19]$$

with:

$${}^i\mathbf{L}_{j,n} = {}^i\mathbf{R}_j {}^j\mathbf{P}_n \quad [5.20]$$

The k^{th} column of ${}^i\mathbf{J}_{n,j}$, deduced from equation [5.17], is expressed in frame \mathbf{R}_i as:

$${}^i\mathbf{J}_{n,j}(:,k) = \begin{bmatrix} \sigma_k {}^i\mathbf{a}_k + \bar{\sigma}_k (-{}^k\mathbf{P}_{j_y} {}^i\mathbf{s}_k + {}^k\mathbf{P}_{j_x} {}^i\mathbf{n}_k) \\ \bar{\sigma}_k {}^i\mathbf{a}_k \end{bmatrix} \quad [5.21]$$

We note that ${}^i\mathbf{J}_n = {}^i\mathbf{J}_{n,n}$. Thus, the matrix ${}^s\mathbf{J}_n$ can be expressed by the multiplication of the following three matrices where the first two are of full-rank:

$${}^s\mathbf{J}_n = \begin{bmatrix} {}^s\mathbf{R}_i & \mathbf{0}_3 \\ \mathbf{0}_3 & {}^s\mathbf{R}_i \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & -\hat{\mathbf{L}}_{j,n} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} {}^i\mathbf{J}_{n,j} \quad [5.22]$$

In general, the shift frame R_j and the projection frame R_i leading to a simple matrix ${}^i\mathbf{J}_{n,j}$ are chosen such that $i = \text{integer}(n/2)$ and $j = i + 1$.

Thus, for a six degree-of-freedom robot, the simplest Jacobian matrix is ${}^3\mathbf{J}_{6,4}$. If the robot has a spherical wrist, the vector $\mathbf{L}_{4,6}$ is zero and consequently ${}^3\mathbf{J}_{6,4} = {}^3\mathbf{J}_6$.

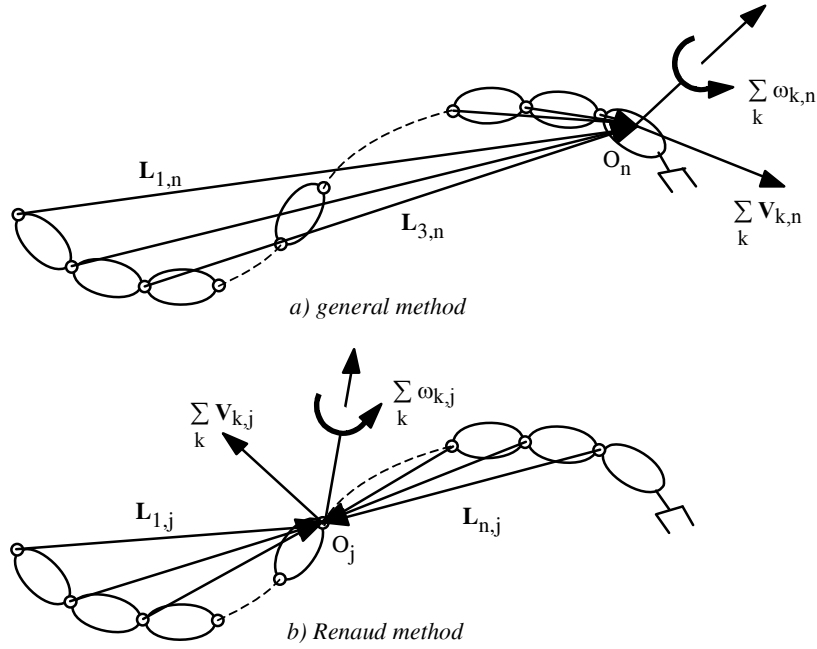


Figure 5.4. Principle of Renaud method

5.5. Efficient computation of the end-effector velocity

Having calculated \mathbf{J}_n , the linear and angular velocities \mathbf{v}_n and $\boldsymbol{\omega}_n$ of frame R_n can be obtained from equation [5.4a]. However, in order to reduce the computational

cost, it is more efficient, as we will see in Chapter 9, to use the following recursive equations for $j = 1, \dots, n$:

$$\begin{aligned} {}^j\omega_{j-1} &= {}^jR_{j-1} {}^{j-1}\omega_{j-1} \\ {}^j\omega_{j-1} &= {}^jR_{j-1} {}^{j-1}\omega_{j-1} \\ {}^jv_j &= {}^jR_{j-1} ({}^{j-1}v_{j-1} + {}^{j-1}\omega_{j-1} \times {}^{j-1}p_j) + \sigma_j \dot{q}_j {}^j\mathbf{a}_j \end{aligned} \quad [5.23]$$

where ${}^j\mathbf{a}_j$ is the unit vector $[0 \ 0 \ 1]^T$. We initialize the algorithm by the linear and angular velocities of the robot base (v_0 and ω_0 respectively), which are zero if the base is fixed.

5.6. Dimension of the task space of a robot

At a given joint configuration \mathbf{q} , the rank r of the Jacobian matrix iJ_n – hereafter written as \mathbf{J} to simplify the notation – corresponds to the number of degrees of freedom of the end-effector. It defines the dimension of the accessible task space at this configuration. The number of degrees of freedom of the task space of a robot, M , is equal to the maximum rank, r_{\max} , which the Jacobian matrix can have at all possible configurations. Noting the number of degrees of freedom of the robot as N (equal to n for serial robots), the following cases are considered [Gorla 84]:

- if $N = M$, the robot is non-redundant and has just the number of joints required to provide M degrees of freedom to the end-effector;
- if $N > M$, the robot is redundant of order $(N - M)$. It has more joints than required to provide M degrees of freedom to the end-effector;
- if $r < M$, the Jacobian matrix is rank deficient. The robot is at a singular configuration of order $(M - r)$. At this configuration, the robot cannot generate end-effector velocity along some directions of the task space, which are known as *degenerate directions*. When the matrix \mathbf{J} is square, the singularities are obtained by the zeros of $\det(\mathbf{J}) = 0$, where $\det(\mathbf{J})$ indicates the determinant of the Jacobian matrix of the robot. They correspond to the zeros of $\det(\mathbf{J}\mathbf{J}^T) = 0$ for redundant robots.

• **Example 5.4.** Computation of the singularities of the Stäubli RX-90 robot. Noting that the Jacobian matrix 3J_6 (obtained in Example 5.3) has the following particular form:

$${}^3J_6 = \begin{bmatrix} \mathbf{A} & \mathbf{0}_3 \\ \mathbf{B} & \mathbf{C} \end{bmatrix}$$

we obtain $\det({}^3\mathbf{J}_6) = \det(\mathbf{A}) \det(\mathbf{C}) = -C_3 D_3 RL_4 S_5 (S_{23} RL_4 - C_2 D_3)$.

The maximum rank is $r_{\max} = 6$. The robot is not redundant because it has six degrees of freedom. However, this rank drops to five in the following three singular configurations:

$$\begin{cases} C_3 = 0 \\ S_{23}RL_4 - C_2D_3 = 0 \\ S_5 = 0 \end{cases}$$

- when $C_3 = 0$, the robot is fully extended (Figure 4.2c) or fully folded. In this case, the origin O_6 is located on the boundary of its workspace: this elbow singularity has not been deduced from the inverse geometric model (§ 4.3.2, Example 4.1). In this configuration, where the second row of ${}^3\mathbf{J}_6$ is zero, the robot cannot generate linear velocity for O_6 along the direction O_6O_2 ;
- the singularity $S_{23}RL_4 - C_2D_3 = 0$ (Figure 4.2a), already deduced from the inverse geometric model, corresponds to a configuration in which O_6 is located on the \mathbf{z}_0 axis (shoulder singularity). In this configuration, where $P_x = P_y = 0$, the third row of ${}^3\mathbf{J}_6$ is zero. The robot cannot generate velocity for O_6 along the normal to the plane containing the points O_2 , O_3 and O_6 ;
- for $S_5 = 0$ (Figure 4.2b), the axes of the joints θ_4 and θ_6 are aligned, resulting in the loss of one degree of freedom of the robot. We notice that the columns 4 and 6 of ${}^3\mathbf{J}_6$ are identical. In this singular configuration (wrist singularity) the degenerate direction depends on the values of the other joints, it can be determined using the singular values decomposition (section 5.8.1). This wrist singularity has already been deduced from the inverse geometric model.

5.7. Analysis of the robot workspace

The analysis of the workspace is very important for the design, selection and programming of robots.

5.7.1. Workspace

Let $\mathbf{q} = [q_1, \dots, q_n]$ be an element of the joint space and let $\mathbf{X} = [x_1, \dots, x_m]$ be the corresponding element in the task space, such that:

$$\mathbf{X} = \mathbf{f}(\mathbf{q}) \quad [5.24]$$

The joint domain \mathbf{Q} is defined as the set of all reachable configurations taking into account the joint limits:

$$\mathbf{Q} = \{\mathbf{q} \mid q_{i\min} \leq q_i \leq q_{i\max}, \forall i = 1, \dots, n\} \quad [5.25]$$

The image of \mathbf{Q} by the direct geometric model defines the workspace W of the robot:

$$W = \mathbf{f}(\mathbf{Q}) \quad [5.26]$$

Thus, the workspace W is the set of positions and orientations reachable by the robot end-effector. Its geometry depends on the robot architecture. Its boundaries are defined by the singularities and the joint limits. However, when there is an obstacle in the robot workspace, additional boundaries limiting the reachable zones will appear [Wenger 89].

For robots with two joints, the workspace is easy to obtain and can be visualized in a plane. For a three degree-of-freedom positioning shoulder, the workspace can be represented by a generic planar cross section of W . This cross section contains the axis of the first joint if it is revolute, whereas it is perpendicular to the axis of the first joint if it is prismatic. The whole workspace is obtained from the generic cross section by rotating it about (or translating it along) the first joint axis. However, if there are obstacles or joint limits, the generic planar section is not sufficient for a complete analysis of the workspace.

In general, the workspace is a 6-dimensional space, which is difficult to handle. However, we can study its projection in the 3-dimensional position space.

5.7.2. Singularity branches

The *singularity branches* are the connected components of the set of singular configurations of \mathbf{Q} . Since the singularities are always independent of the first joint, we can represent them in the joint space excluding the first joint. They are represented by surfaces of \mathbf{Q} . However, for some particular cases, they can be reduced to subspaces of fewer dimensions (curves or points for example), which do not have a boundary in \mathbf{Q} .

For the two degree-of-freedom planar robot with revolute joints shown in Figure 5.5, the determinant of the Jacobian matrix is equal to $L_1 L_2 S_2$. The singularity branches, assuming unlimited joint ranges, are defined by the lines $\theta_2 = 0$ and $\theta_2 = \pm \pi$ (Figure 5.6). The corresponding workspace is presented in Figure 5.7.

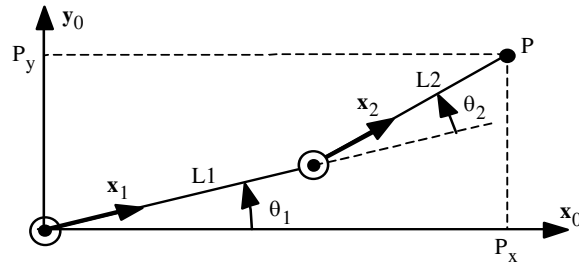


Figure 5.5. Two degree-of-freedom planar robot

For the Stäubli RX-90 robot, the joint space is partitioned by three singularity surfaces $C3 = 0$, $S23 \text{ RL4} - C2 \text{ D3} = 0$ and $S5 = 0$. Figure 5.8 shows these surfaces in the $(\theta_2, \theta_3, \theta_5)$ space and in the (θ_2, θ_3) plane.

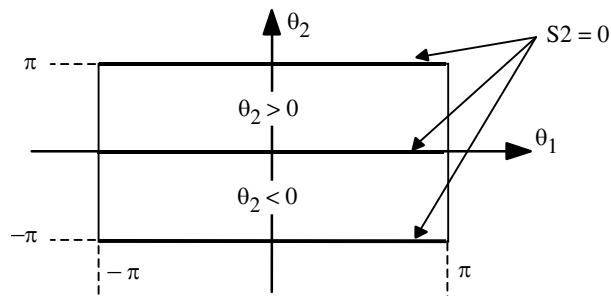


Figure 5.6. Singularity branches of the planar robot with unlimited joints

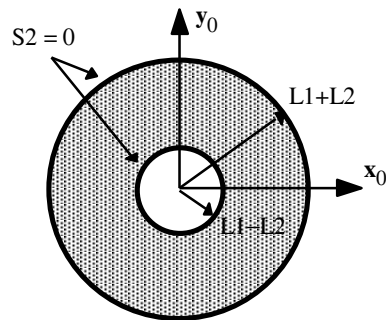
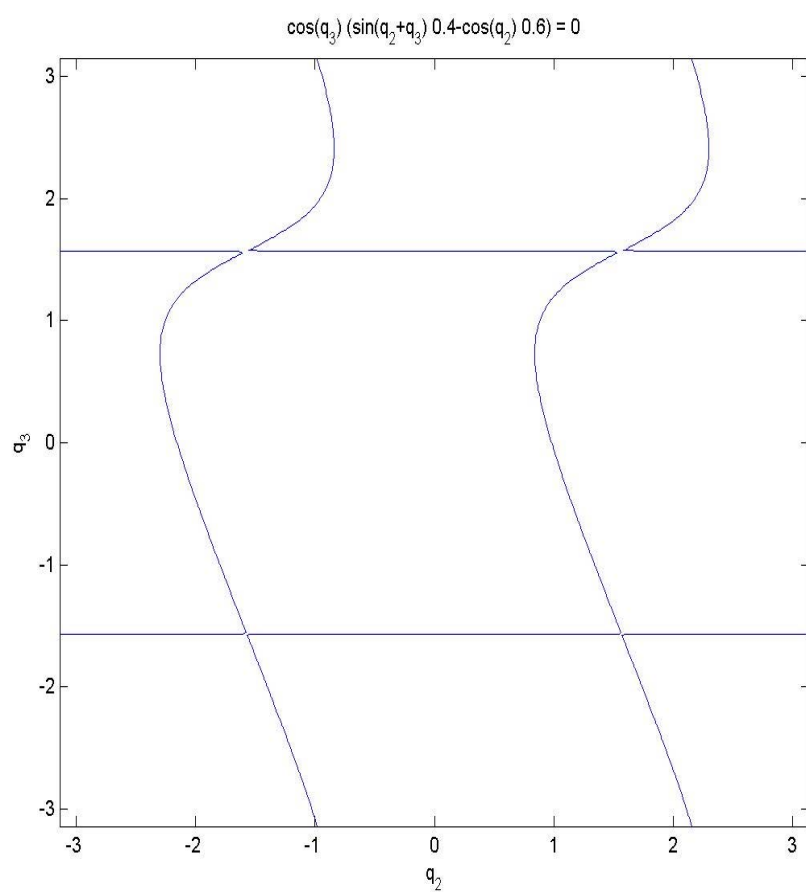
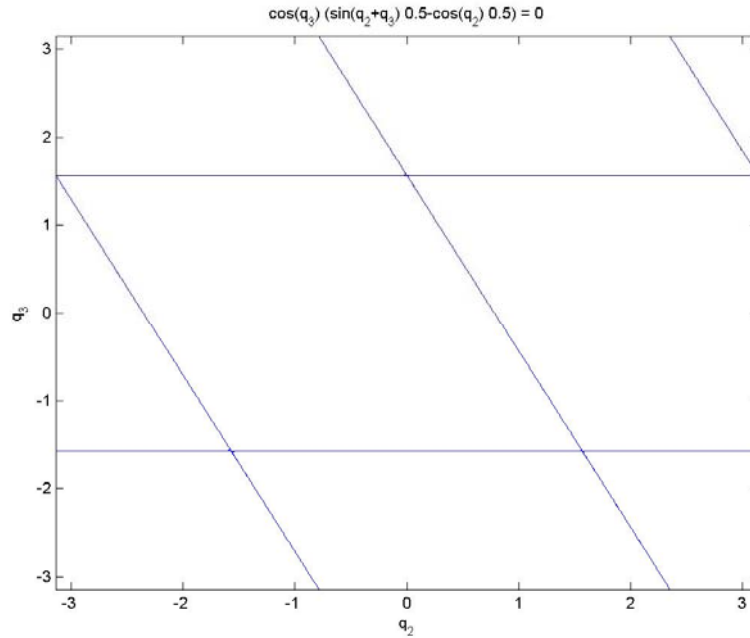


Figure 5.7. Workspace of the planar robot with unlimited joints ($L1 > L2$)





5.7.3. Jacobian surfaces

Mapping the singularities into the workspace generally leads to surfaces (or subspaces with fewer dimensions) called *Jacobian surfaces*. These surfaces divide W into regions where the number of solutions of the IGM is constant and even [Roth 76], [Kholi 85], [Burdick 88]. In the presence of joint limits, additional boundaries appear in W , which define new regions in which the number of solutions of the IGM is always constant but not necessarily even [Spanos 85]. The Jacobian surfaces can be defined as the set of points in W where the IGM has at least two identical solutions [Kholi 87], [Spanos 85]. When the robot has three identical solutions for a point of the Jacobian surface, the robot is said to be *cuspidal* [El Omri 96].

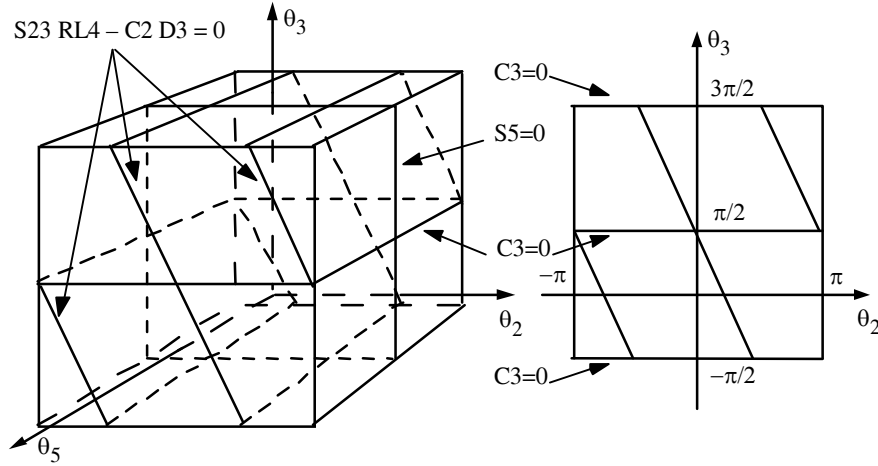


Figure 5.8. Singularity branches of the Stäubli RX-90 robot

In the case of a three degree-of-freedom robot, if the Jacobian surfaces are subspaces of fewer dimensions (for example a curve or a set of isolated points), the IGM for these points has an infinite number of solutions.

For the two degree-of-freedom planar robot shown in Figure 5.5, the Jacobian surfaces correspond to the singular configurations "extended arm" and "folded arm". They are represented by the circles with radii $L_1 + L_2$ and $L_1 - L_2$ respectively (Figure 5.7).

For the anthropomorphic shoulder of the Stäubli RX-90 robot, the Jacobian surfaces in the position workspace are of two types (Figure 5.9). The first is associated with the singular configurations where the point O_6 lies on the axis of the first joint. Their reciprocal mapping in \mathbf{Q} give the singularity surfaces defined by $S_{23}RL_4 - C_2D_3 = 0$. For any point of these configurations, the IGM has an infinite number of solutions since θ_1 can be chosen arbitrarily. The other type of Jacobian surface corresponds to the singular configuration $C_3 = 0$, and is represented by the surfaces of the spheres whose center is O_0 , with radii $D_3 + RL_4$ ("extended arm" configuration) and $D_3 - RL_4$ ("folded arm" configuration) defining the external and internal boundaries of the workspace respectively. For the Stäubli RX-90 robot, the internal sphere is reduced to a point because $D_3 = RL_4$.

5.7.4. Concept of aspect

The concept of aspect has been introduced by Borrel [Borrel 86]. The aspects are the connected regions of the joint space inside which no minor of order M extracted from the Jacobian matrix \mathbf{J} is zero, except if this minor is zero everywhere in the joint domain. For a non-redundant robot manipulator, the only minor of order M is

the Jacobian matrix itself. Therefore, the aspects are limited by the singularity branches and the joint limits (Figures 5.6 and 5.8). Consequently, they represent the maximum singularity-free regions.

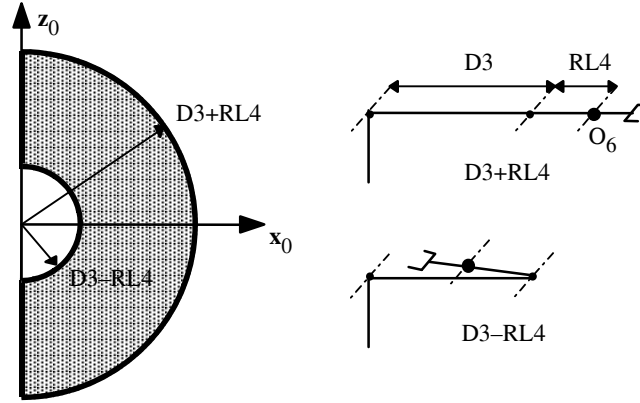


Figure 5.9. *Generic section of the workspace of an anthropomorphic shoulder with unlimited joints*

For a long time, it has been thought that the aspects also represent the uniqueness domains of the IGM solutions. Although this is indeed the case for most industrial robots with simple architectures, which are classified as non-cuspidal robots [El Omri 96], the IGM of cuspidal robots can have several solutions in the same aspect. Thus, a cuspidal robot can move from one IGM solution to another without encountering a singularity. Figure 5.10 shows a cuspidal robot with three revolute joints whose successive axes are perpendicular. The inverse geometric solution of the point X (Figure 5.11a) whose coordinates are $P_x = 2.5$, $P_y = 0$, $P_z = 0.6$ is given by the following four configurations (in degrees):

$$\mathbf{q}^1 = [-101.52 \quad -158.19 \quad 104.88]^T,$$

$$\mathbf{q}^2 = [-50.92 \quad -46.17 \quad 141.16]^T$$

$$\mathbf{q}^3 = [-164.56 \quad -170.02 \quad -12.89]^T,$$

$$\mathbf{q}^4 = [10.13 \quad -22.33 \quad -106.28]^T$$

The joint space of this robot is divided into two aspects (Figure 5.11a). We notice that the configurations \mathbf{q}^2 and \mathbf{q}^3 are located in the same aspect whereas \mathbf{q}^1 and \mathbf{q}^4 fall in the other aspect.

For cuspidal robots, the uniqueness domains of the IGM in the joint space are separated by the *characteristic surfaces* [Wenger 92], which are defined as the

mapping of the Jacobian surfaces in the joint space using the IGM. Figure 5.11b shows the singularities and the characteristic surfaces of the shoulder structure of Figure 5.10.

There is no general simple rule to identify the architectures of non-cuspidal robots. However, Table 5.1 gives a list of non-cuspidal shoulders as presented in [Wenger 93], [Wenger 98].

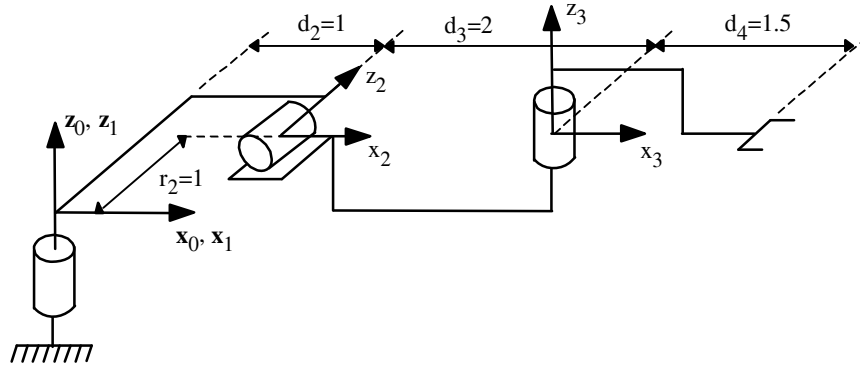


Figure 5.10. Example of a cuspidal shoulder [Wenger 92]

Table 5.1. Non-cuspidal shoulders

PPP	RPP	PRP	PPR	RRR	PRR	RPR	RRP
all	all	all	all	$s\alpha_2=0$ $s\alpha_3=0$ $d_2=0$ $d_3=0$ $(c\alpha_2=0, r_2=0$ and $r_3=0)$	$c\alpha_2=0$ $s\alpha_3=0$ $d_3=0$ $(s\alpha_2=0$ and $r_3=0)$ $(s\alpha_2=0$ and $c\alpha_3=0)$	$c\alpha_2=0$ $c\alpha_3=0$ $s_2=0$ $d_3+d_2c_2=0$	$s\alpha_2=0$ $c\alpha_3=0$ $d_2=0$ $(c\alpha_2=0, s\alpha_3=0$ and $r_2=0)$ $d_3+d_4c_3=0$

5.7.5. *t*-connected subspaces

The *t*-connected subspaces are the regions in the workspace where any continuous trajectory can be followed by the robot end-effector. These subspaces are the mapping of the uniqueness domains into W using the DGM. For the non-cuspidal robots, the largest *t*-connected subspaces are the mapping of the aspects (and more generally of the free connected regions of the aspects when the environment is cluttered with obstacles [Wenger 89]). We do not present here the

definition of the t-connected subspaces for the cuspidal robots. The interested reader can refer to [El Omri 96].

For the two degree-of-freedom planar robot shown in Figure 5.5, the straight line $S_2 = 0$ separates the joint space domain into two aspects (Figure 5.12a) corresponding to the two solutions of the IGM, $\theta_2 > 0$ and $\theta_2 < 0$.

The mapping of these aspects in the workspace is identical if the joint ranges are equal to 2π . Figure 5.12b shows, for certain joint limits θ_{\max} and θ_{\min} , the t-connected regions: the hatched and non-hatched zones represent the mapping of the aspects $\theta_2 > 0$ and $\theta_2 < 0$ respectively. The trajectory PP' is located in the region mapped by the aspect $\theta_2 < 0$: thus it can only be realized if the initial configuration of the robot is $\theta_2 < 0$. Otherwise, one of the joints reaches its limit before arriving at the final position.

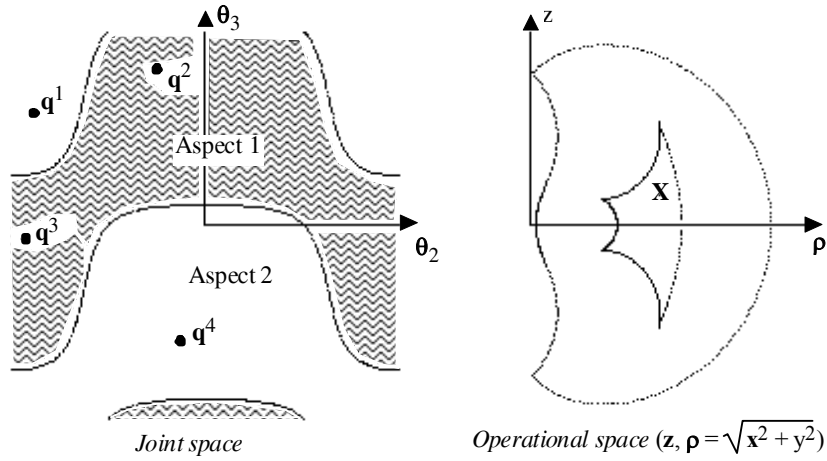


Figure 5.11a. Aspects and workspace of the cuspidal shoulder of Figure 5.10

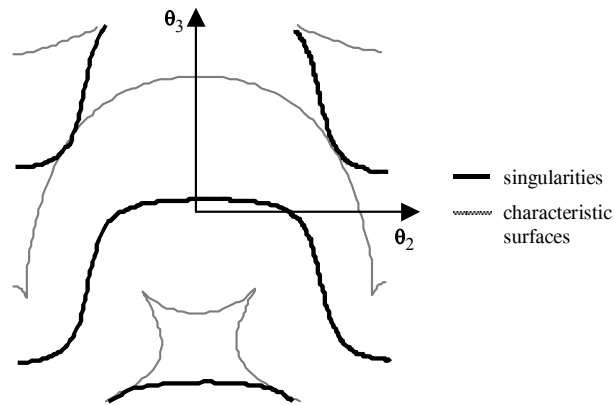


Figure 5.11b. Singularity branches and characteristic surfaces of the cuspidal shoulder

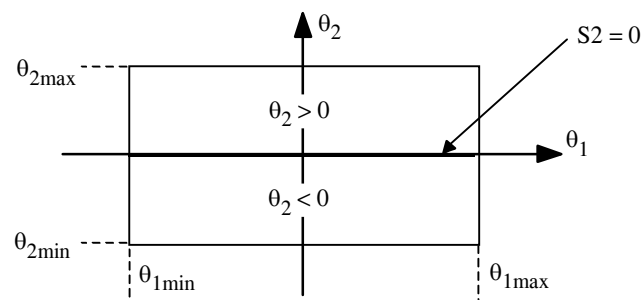


Figure 5.12a. Aspects in the presence of joint limits

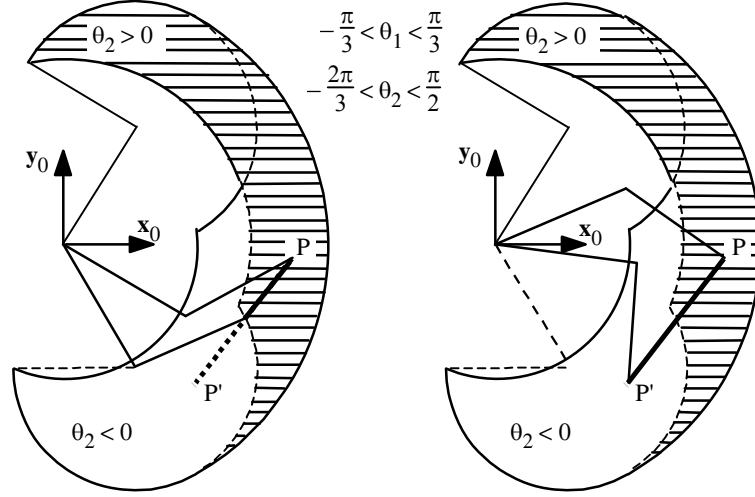


Figure 5.12b. *t*-connected regions in the workspace

5.8. Velocity transmission between joint space and task space

5.8.1. Singular value decomposition

At a given configuration, the $(m \times n)$ matrix \mathbf{J} represents a linear mapping of the joint space velocities into the task space velocities. For simplicity, we write the kinematic Jacobian matrix \mathbf{J}_n as \mathbf{J} . When the end-effector coordinates are independent, we have $n=N$ and $m=M$.

The singular value decomposition (SVD) theory states that for any $(m \times n)$ matrix \mathbf{J} of rank r [Lawson 74], [Dongarra 79], [Klema 80], there exist orthogonal matrices \mathbf{U} and \mathbf{V} of dimensions $(m \times m)$ and $(n \times n)$ respectively such that:

$$\mathbf{J} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad [5.27]$$

The $(m \times n)$ matrix $\mathbf{\Sigma}$ has the following form:

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{S}_{r \times r} & \mathbf{0}_{r \times (n-r)} \\ \mathbf{0}_{(m-r) \times r} & \mathbf{0}_{(m-r) \times (n-r)} \end{bmatrix} \quad [5.28]$$

\mathbf{S} is an $(r \times r)$ diagonal matrix, formed by the non-zero singular values of \mathbf{J} , which are arranged in decreasing order such that $s_1 \geq s_2 \geq \dots \geq s_r$. The singular values of \mathbf{J} are the square roots of the eigenvalues of the matrix $\mathbf{J}^T \mathbf{J}$ if $n \geq m$ (or $\mathbf{J} \mathbf{J}^T$ if $n \leq m$). The columns of \mathbf{V} are the eigenvectors of $\mathbf{J}^T \mathbf{J}$ and are called *right singular vectors* or *input vectors* of \mathbf{J} . The columns of \mathbf{U} are the eigenvectors of $\mathbf{J} \mathbf{J}^T$ and are called *left singular vectors* or *output vectors*.

Using equation [5.27], the kinematic model becomes:

$$\dot{\mathbf{X}} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \dot{\mathbf{q}} \quad [5.29]$$

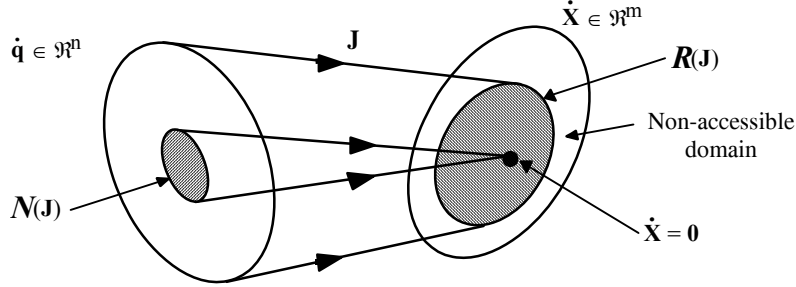
Since $s_i = 0$ for $i > r$, we can write:

$$\dot{\mathbf{X}} = \sum_{i=1}^r s_i \mathbf{U}_i \mathbf{V}_i^T \dot{\mathbf{q}} \quad [5.30]$$

From equation [5.30], we deduce that (Figure 5.13):

- the vectors $\mathbf{V}_1, \dots, \mathbf{V}_r$ form an orthonormal basis for the subspace of $\dot{\mathbf{q}}$ generating an end-effector velocity;
- the vectors $\mathbf{V}_{r+1}, \dots, \mathbf{V}_n$ form an orthonormal basis for the subspace of $\dot{\mathbf{q}}$ giving $\dot{\mathbf{X}} = 0$. In other words, they define the null space of \mathbf{J} , denoted by $\mathcal{N}(\mathbf{J})$;
- the vectors $\mathbf{U}_1, \dots, \mathbf{U}_r$ form an orthonormal basis for the set of the achievable end-effector velocities $\dot{\mathbf{X}}$. Hence, they define the range space of \mathbf{J} , denoted by $\mathcal{R}(\mathbf{J})$;
- the vectors $\mathbf{U}_{r+1}, \dots, \mathbf{U}_m$ form an orthonormal basis for the subspace composed of the set of $\dot{\mathbf{X}}$ that cannot be generated by the robot. In other words, they define the complement of the range space, denoted by $\mathcal{R}(\mathbf{J})^\perp$;
- the singular values represent the velocity transmission ratio from the joint space to the task space. In fact, multiplying equation [5.30] by \mathbf{U}_i^T yields:

$$\mathbf{U}_i^T \dot{\mathbf{X}} = s_i \mathbf{V}_i^T \dot{\mathbf{q}} \quad \text{for } i = 1, \dots, r \quad [5.31]$$

Figure 5.13. Null space and range space of \mathbf{J} (from [Asada 86])

– since $\mathbf{J}^T = \mathbf{V} \mathbf{\Sigma} \mathbf{U}^T$, we deduce that:

$$\begin{aligned}\mathfrak{R}^m &= \mathcal{R}(\mathbf{J}) + \mathcal{R}(\mathbf{J})^\perp = \mathcal{R}(\mathbf{J}) + \mathcal{N}(\mathbf{J}^T) \\ \mathfrak{R}^n &= \mathcal{R}(\mathbf{J}^T) + \mathcal{N}(\mathbf{J})\end{aligned}$$

5.8.2. Velocity ellipsoid: velocity transmission performance

The velocity transmission performance of a mechanism can be evaluated through the kinematic model [5.1]. Let us suppose that the joint velocities are limited such that:

$$-\dot{\mathbf{q}}_{\max} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{\max} \quad [5.32]$$

At a given configuration \mathbf{q} , the task space velocity satisfying these conditions belongs to:

$$\dot{\mathbf{X}}_{\min} \leq \dot{\mathbf{X}} \leq \dot{\mathbf{X}}_{\max} \quad [5.33]$$

with:

$$\dot{\mathbf{X}}_{\max} = \max(\mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}) \quad [5.34]$$

$$\dot{\mathbf{X}}_{\min} = \min(\mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}) \quad [5.35]$$

Thus, the set of possible joint velocities (equation [5.32]) can be represented geometrically by a hyper-parallelepiped in the joint space. Equation [5.33] can also be represented by a hyper-parallelepiped in the task space. In this section, we develop another common approach to studying the velocity transmission between the joint space and the task space using an analytical ellipsoidal representation.

Let us consider the joint velocities contained in the unit sphere of the joint velocity space, such that [Yoshikawa 84b]:

$$\dot{\mathbf{q}}^T \dot{\mathbf{q}} \leq 1 \quad [5.36]$$

We can show that the corresponding velocities in the task space are defined by the ellipsoid:

$$\dot{\mathbf{X}}^T (\mathbf{J} \mathbf{J}^T)^{-1} \dot{\mathbf{X}} \leq 1 \quad [5.37]$$

The velocity ellipsoid is a useful tool for analyzing the velocity transmission performance of a robot at a given configuration. It is called the *manipulability ellipsoid*. The principal axes of the ellipsoid are given by the vectors $\mathbf{U}_1, \dots, \mathbf{U}_m$, which are the eigenvectors of $\mathbf{J} \mathbf{J}^T$. The lengths of the principal axes are determined by the singular values s_1, \dots, s_m of \mathbf{J} . The optimum direction to generate velocity is along the major axis where the transmission ratio is maximum. Conversely, the velocity is most accurately controlled along the minor axis. Figure 5.14 shows the velocity ellipsoid for a 2R planar mechanism.

The volume of the velocity ellipsoid of a robot gives a measurement of its capacity to generate velocity. Consequently, we define the velocity manipulability of a robot as:

$$w(\mathbf{q}) = \sqrt{\det(\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}))} \quad [5.38]$$

For a non-redundant robot, this expression becomes:

$$w(\mathbf{q}) = |\det[\mathbf{J}(\mathbf{q})]| \quad [5.39]$$

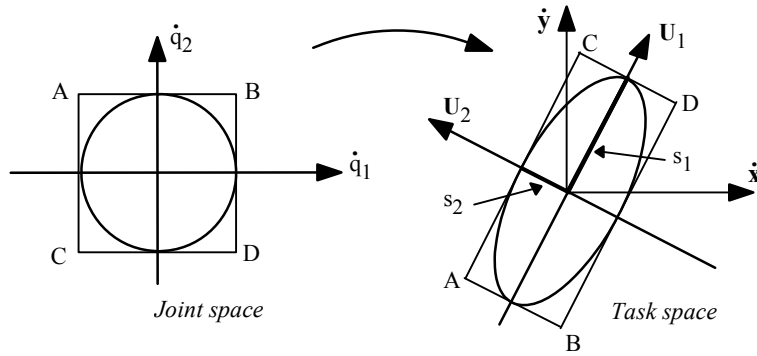


Figure 5.14. Velocity ellipsoid for a two degree-of-freedom planar robot

5.9. Static model

In this section, we establish the static model, which provides the joint torques (for revolute joints) or forces (for prismatic joints) corresponding to the wrench (forces and moments) exerted by the end-effector on the environment. We also discuss the duality between the kinematic model and the static model.

5.9.1. Representation of a wrench

Let us recall (§ 2.6) that a wrench \mathbf{F}_i is represented by the screw, which is composed of a force \mathbf{f}_i and a moment \mathbf{m}_i :

$$\mathbf{F}_i = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{m}_i \end{bmatrix} \quad [5.40]$$

We assume, unless otherwise stated, that the moment is defined about the point O_i , origin of frame R_i . Let the static wrench \mathbf{F}_{en} to be exerted on the environment be defined as:

$$\mathbf{F}_{en} = \begin{bmatrix} \mathbf{f}_{en} \\ \mathbf{m}_{en} \end{bmatrix} = \begin{bmatrix} f_x & f_y & f_z & m_x & m_y & m_z \end{bmatrix}^T \quad [5.41]$$

The subscript n indicates that the wrench is expressed at the origin O_n of frame R_n .

5.9.2. Mapping of an external wrench into joint torques

To compute the joint torques and forces Γ_e of a serial robot such that its end-effector can exert a static wrench \mathbf{F}_{en} , we make use of the principle of virtual work, which states that:

$$\Gamma_{en}^T \mathbf{dq}^* = \mathbf{F}_{en}^T \begin{bmatrix} d\mathbf{P}_n^* \\ \delta_n^* \end{bmatrix} \quad [5.42]$$

where the superscript (*) indicates virtual displacements.

Substituting $d\mathbf{P}_n^*$ and δ_n^* from equation [5.4b] gives:

$$\mathbf{\Gamma}_e = \mathbf{J}_n^T \mathbf{F}_{en} \quad [5.43]$$

We can use either the Jacobian matrix ${}^n\mathbf{J}_n$ or ${}^0\mathbf{J}_n$ depending on whether the wrench \mathbf{F}_{en} is referred to frame R_n or frame R_0 respectively.

5.9.3. Velocity-force duality

The Jacobian matrix appearing in the static model (equation [5.43]) is the same as that used in the differential or kinematic model. By analogy with the velocity transmission analysis (§ 5.8.1), we deduce the following results (Figure 5.15) [Asada 86]:

- the torques of the actuators are uniquely determined for an arbitrary wrench \mathbf{f} ; the range space of \mathbf{J}^T , denoted as $\mathcal{R}(\mathbf{J}^T)$, is the set of $\mathbf{\Gamma}$ balancing the static wrench \mathbf{f} according to equation [5.43];
- for a zero $\mathbf{\Gamma}$, the corresponding static wrench can be non-zero; we thus define the null space of \mathbf{J}^T , $\mathcal{N}(\mathbf{J}^T)$, as the set of static wrenches that do not require actuator torques in order to be balanced. In this case, the endpoint wrench is borne by the structure of the robot. Note that the null space of \mathbf{J}^T , $\mathcal{N}(\mathbf{J}^T)$, which is the orthogonal complement of $\mathcal{R}(\mathbf{J})$, also represents the set of directions along which the robot cannot generate velocity;
- some joint torques $\mathbf{\Gamma}$ cannot be compensated by \mathbf{f} . These torques correspond to the vectors of the null space $\mathcal{N}(\mathbf{J})$, orthogonal complement of the space $\mathcal{R}(\mathbf{J}^T)$.

The basis of these spaces can be defined using the columns of the matrices \mathbf{U} and \mathbf{V} of the singular value decomposition of \mathbf{J} as indicated for the velocity case (§ 5.8.1).

Analogously, we can study the force transmission performance using a force manipulability ellipsoid, which corresponds to the set of achievable wrench in the task space \mathfrak{R}^m corresponding to the constraint $\mathbf{\Gamma}^T \mathbf{\Gamma} \leq 1$. Thus, the force ellipsoid is defined by $\mathbf{F}^T \mathbf{J} \mathbf{J}^T \mathbf{F} \leq 1$. Consequently, we can deduce that the velocity ellipsoid (equation [5.37]) and the force ellipsoid have the same principal axes but the axis lengths are reciprocal (Figure 5.16). This means that the optimum direction for generating velocity is the optimum direction for controlling force. Similarly, the optimal direction for exerting force is also the optimum direction for controlling velocity.

From the control point of view, this behavior makes sense: the velocity is controlled most accurately in the direction where the robot can resist large force

disturbances, and force is most accurately controlled in the direction where the robot can rapidly adapt its motion.

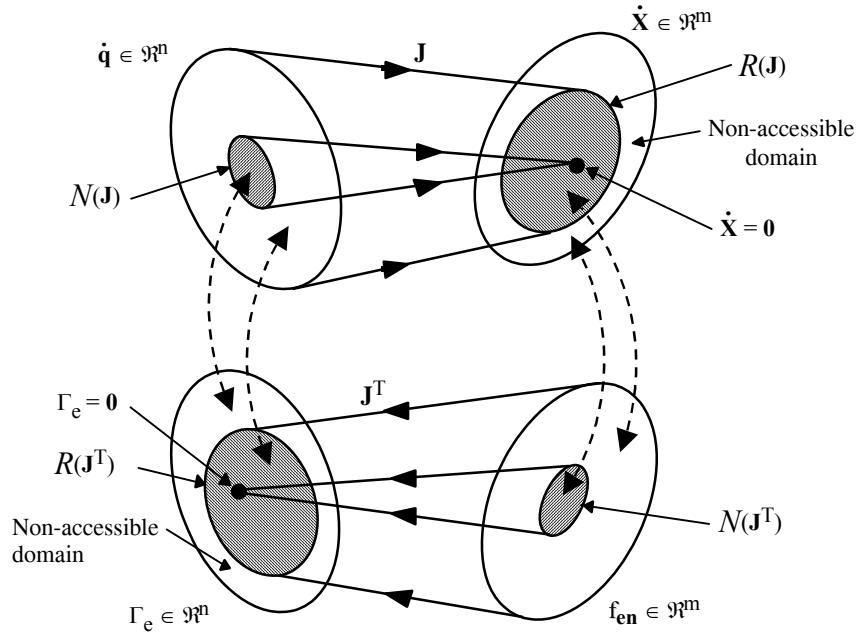


Figure 5.15. Velocity-force duality (from [Asada 86])

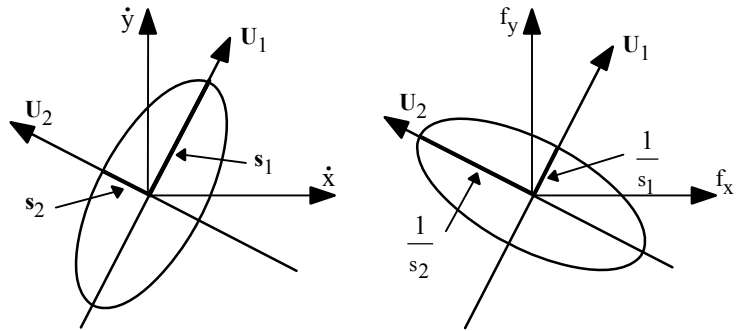


Figure 5.16. Velocity and force ellipsoids

5.10. Second order kinematic model

The second order kinematic model allows us to compute the acceleration of the end-effector in terms of positions, velocities and accelerations of the joints. By differentiating equation [5.1] with respect to time, we obtain the following expression:

$$\ddot{\mathbf{X}} = \mathbf{J} \ddot{\mathbf{q}} + \dot{\mathbf{J}} \dot{\mathbf{q}} \quad [5.44]$$

where:

$$\dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{d}{dt} \mathbf{J}(\mathbf{q}) \quad [5.45]$$

Using the kinematic Jacobian matrix, the second order kinematic model can be written as:

$$\dot{\mathbf{V}}_n = \begin{bmatrix} \dot{\mathbf{v}}_n \\ \dot{\boldsymbol{\omega}}_n \end{bmatrix} = \mathbf{J}_n \ddot{\mathbf{q}} + \dot{\mathbf{J}}_n \dot{\mathbf{q}} \quad [5.46]$$

However, it is most efficient from the computational cost point of view to obtain $\dot{\mathbf{v}}_n$ and $\dot{\boldsymbol{\omega}}_n$ from the following recursive equations, for $j = 1, \dots, n$, which will be developed in Chapter 9:

$$\begin{cases} {}^j\dot{\boldsymbol{\omega}}_j = {}^j\mathbf{R}_{j-1} {}^{j-1}\dot{\boldsymbol{\omega}}_{j-1} + \bar{\sigma}_j (\ddot{q}_j {}^j\mathbf{a}_j + {}^j\boldsymbol{\omega}_{j-1} \times \dot{q}_j {}^j\mathbf{a}_j) \\ {}^j\mathbf{U}_j = {}^j\hat{\boldsymbol{\omega}}_j + {}^j\hat{\boldsymbol{\omega}}_j {}^j\hat{\boldsymbol{\omega}}_j \\ {}^j\dot{\mathbf{v}}_j = {}^j\mathbf{R}_{j-1} ({}^{j-1}\dot{\mathbf{v}}_{j-1} + {}^{j-1}\mathbf{U}_{j-1} {}^{j-1}\mathbf{P}_j) + \sigma_j (\ddot{q}_j {}^j\mathbf{a}_j + 2{}^j\boldsymbol{\omega}_{j-1} \times \dot{q}_j {}^j\mathbf{a}_j) \end{cases} \quad [5.47]$$

The angular velocities ${}^j\boldsymbol{\omega}_{j-1}$ and ${}^j\boldsymbol{\omega}_j$ are calculated using equation [5.23].

In certain applications, such as the control in the task space (§ 14.4.3), we need to compute the vector $\dot{\mathbf{J}} \dot{\mathbf{q}}$. Instead of taking the derivative of \mathbf{J} with respect to time and multiplying by $\dot{\mathbf{q}}$, it is more efficient to make use of the recursive equations [5.47] with $\ddot{\mathbf{q}}$ equal to zero in order to leave out the terms involving $\ddot{\mathbf{q}}$ [Khalil 87a].

5.11. Kinematic model associated with the task coordinates representation

Let $\mathbf{X} = \begin{bmatrix} \mathbf{X}_p \\ \mathbf{X}_r \end{bmatrix}$ be any representation of the location of frame R_n relative to frame R_0 , where \mathbf{X}_p and \mathbf{X}_r denote the position and orientation vectors respectively. The relationships between the velocities $\dot{\mathbf{X}}_p$ and $\dot{\mathbf{X}}_r$ and the velocities ${}^0\mathbf{v}_n$ and ${}^0\boldsymbol{\omega}_n$ of frame R_n are given as:

$$\begin{cases} \dot{\mathbf{X}}_p = \boldsymbol{\Omega}_p {}^0\mathbf{v}_n \\ \dot{\mathbf{X}}_r = \boldsymbol{\Omega}_r {}^0\boldsymbol{\omega}_n \end{cases} \quad [5.48]$$

Similar relations can be derived to express the differential vectors $d\mathbf{X}_p$ and $d\mathbf{X}_r$ as functions of the vectors ${}^0d\mathbf{P}_n$ and ${}^0\delta_n$:

$$\begin{cases} d\mathbf{X}_p = \boldsymbol{\Omega}_p d\mathbf{P}_n \\ d\mathbf{X}_r = \boldsymbol{\Omega}_r \delta_n \end{cases} \quad [5.49]$$

In matrix form, equation [5.48] becomes:

$$\begin{bmatrix} \dot{\mathbf{X}}_p \\ \dot{\mathbf{X}}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega}_p & \mathbf{0}_3 \\ \mathbf{0}_3 & \boldsymbol{\Omega}_r \end{bmatrix} \begin{bmatrix} {}^0\mathbf{v}_n \\ {}^0\boldsymbol{\omega}_n \end{bmatrix} = \boldsymbol{\Omega} \begin{bmatrix} {}^0\mathbf{v}_n \\ {}^0\boldsymbol{\omega}_n \end{bmatrix} \quad [5.50]$$

Using equation [5.4a], we deduce that:

$$\begin{bmatrix} \dot{\mathbf{X}}_p \\ \dot{\mathbf{X}}_r \end{bmatrix} = \boldsymbol{\Omega} {}^0\mathbf{J}_n \dot{\mathbf{q}} = \mathbf{J}_x \dot{\mathbf{q}} \quad [5.51]$$

with:

$$\mathbf{J}_x = \boldsymbol{\Omega} {}^0\mathbf{J}_n \quad [5.52]$$

The matrix $\boldsymbol{\Omega}_p$ is equal to \mathbf{I}_3 when the position of frame R_n is described by the Cartesian coordinates.

In this section, we show how to calculate $\mathbf{\Omega}_r$ and $\mathbf{\Omega}_r^{-1}$ for different orientation representations. These expressions are necessary for establishing the kinematic model corresponding to the representation at hand. When the orientation description is not redundant, the inverse of $\mathbf{\Omega}$ can be written as:

$$\mathbf{\Omega}^{-1} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{\Omega}_r^{-1} \end{bmatrix} \quad [5.53]$$

If the description of the orientation is redundant, which is the case with the direction cosines and the quaternions (Euler parameters), the matrices $\mathbf{\Omega}_r$, and consequently $\mathbf{\Omega}$, are rectangular. We then use the so-called left inverse, which is a particular case of the pseudoinverse (Appendix 4). The left inverse is defined by:

$$\mathbf{\Omega}^+ = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{\Omega}_r^+ \end{bmatrix} \quad [5.54]$$

with:

$$\begin{cases} \mathbf{\Omega}^+ = (\mathbf{\Omega}^T \mathbf{\Omega})^{-1} \mathbf{\Omega}^T \\ \mathbf{\Omega}^+ \mathbf{\Omega} = \mathbf{I}_6 \end{cases} \quad [5.55]$$

Such a matrix exists if $\mathbf{\Omega}$ is of rank 6, which means that $\mathbf{\Omega}_r$ is of rank 3.

5.11.1. Direction cosines

The velocity of the vectors \mathbf{s} , \mathbf{n} , \mathbf{a} are given by:

$$\begin{cases} {}^0\dot{\mathbf{s}}_n = {}^0\boldsymbol{\omega}_n \times {}^0\mathbf{s}_n \\ {}^0\dot{\mathbf{n}}_n = {}^0\boldsymbol{\omega}_n \times {}^0\mathbf{n}_n \\ {}^0\dot{\mathbf{a}}_n = {}^0\boldsymbol{\omega}_n \times {}^0\mathbf{a}_n \end{cases} \quad [5.56]$$

Using the vector product operator defined in [2.32], equations [5.56] can be written in the following matrix form [Khatib 80]:

$$\dot{\mathbf{X}}_r = \begin{bmatrix} {}^0\dot{\mathbf{s}}_n \\ {}^0\dot{\mathbf{n}}_n \\ {}^0\dot{\mathbf{a}}_n \end{bmatrix} = \begin{bmatrix} -{}^0\hat{\mathbf{s}}_n \\ -{}^0\hat{\mathbf{n}}_n \\ -{}^0\hat{\mathbf{a}}_n \end{bmatrix} {}^0\boldsymbol{\omega}_n = \boldsymbol{\Omega}_{CD} {}^0\boldsymbol{\omega}_n \quad [5.57]$$

where $\boldsymbol{\Omega}_{CD}$ is a (9x3) matrix. To calculate $\boldsymbol{\Omega}_{CD}^+$, we use the fact that:

$$\boldsymbol{\Omega}_{CD}^T \boldsymbol{\Omega}_{CD} = 2 \mathbf{I}_3 \quad [5.58]$$

Using equation [5.55] and taking into account that the matrices ${}^0\hat{\mathbf{s}}_n$, ${}^0\hat{\mathbf{n}}_n$, ${}^0\hat{\mathbf{a}}_n$ are skew-symmetric, we obtain:

$$\boldsymbol{\Omega}_{CD}^+ = \frac{1}{2} \boldsymbol{\Omega}_{CD}^T = \frac{1}{2} \begin{bmatrix} {}^0\hat{\mathbf{s}}_n & {}^0\hat{\mathbf{n}}_n & {}^0\hat{\mathbf{a}}_n \end{bmatrix} \quad [5.59]$$

Note:

The following relations can be deduced from [5.56]:

$$\begin{aligned} {}^0\dot{\mathbf{R}}_n &= {}^0\hat{\boldsymbol{\omega}}_n {}^0\mathbf{R}_n \\ {}^0\hat{\boldsymbol{\omega}}_n &= {}^0\dot{\mathbf{R}}_n {}^n\mathbf{R}_0 = {}^0\dot{\mathbf{R}}_n {}^0\mathbf{R}_n^T \\ {}^n\hat{\boldsymbol{\omega}}_n &= {}^n\mathbf{R}_0 {}^0\dot{\mathbf{R}}_n = {}^0\mathbf{R}_n^T {}^0\dot{\mathbf{R}}_n \end{aligned}$$

5.11.2. Euler angles

We deduce from § 3.6.1 that ϕ is the rotation angle about $\mathbf{z}_0 = [0 \ 0 \ 1]^T$, θ is the rotation angle about the current \mathbf{x} axis (after applying $\mathbf{rot}(\mathbf{z}, \phi)$) whose unit vector with respect to \mathbf{R}_0 is $[C\phi \ S\phi \ 0]^T$, and ψ is the rotation angle about the current \mathbf{z} axis (after applying $\mathbf{rot}(\mathbf{z}, \phi) \mathbf{rot}(\mathbf{x}, \theta)$) whose unit vector components with respect to \mathbf{R}_0 are $[S\phi S\theta \ -C\phi S\theta \ C\theta]^T$. Thus, the angular velocity of frame \mathbf{R}_n relative to frame \mathbf{R}_0 is given by:

$${}^0\boldsymbol{\omega}_n = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\phi} + \begin{bmatrix} C\phi \\ S\phi \\ 0 \end{bmatrix} \dot{\theta} + \begin{bmatrix} S\phi S\theta \\ -C\phi S\theta \\ C\theta \end{bmatrix} \dot{\psi} \quad [5.60]$$

thus:

$${}^0\boldsymbol{\omega}_n = \begin{bmatrix} 0 & C\phi & S\phi S\theta \\ 0 & S\phi & -C\phi S\theta \\ 1 & 0 & C\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad [5.61]$$

which we identify with:

$${}^0\boldsymbol{\omega}_n = \boldsymbol{\Omega}_{\text{Eul}}^{-1} \dot{\mathbf{X}}_r = \boldsymbol{\Omega}_{\text{Eul}}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad [5.62]$$

By taking the inverse of $\boldsymbol{\Omega}_{\text{Eul}}^{-1}$, we obtain:

$$\boldsymbol{\Omega}_{\text{Eul}} = \begin{bmatrix} -S\phi \cot\theta & C\phi \cot\theta & 1 \\ C\phi & S\phi & 0 \\ S\phi/S\theta & -C\phi/S\theta & 0 \end{bmatrix} \quad [5.63]$$

$\boldsymbol{\Omega}_{\text{Eul}}$ is singular when $S\theta = 0$, as already obtained in § 3.6.1.

5.11.3. Roll-Pitch-Yaw angles

Similarly, we can write:

$${}^0\boldsymbol{\omega}_n = \begin{bmatrix} 0 & -S\phi & C\phi C\theta \\ 0 & C\phi & S\phi C\theta \\ 1 & 0 & -S\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \boldsymbol{\Omega}_{\text{RPY}}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad [5.64]$$

from which we obtain:

$$\boldsymbol{\Omega}_{\text{RPY}} = \begin{bmatrix} C\phi \tan\theta & S\phi \tan\theta & 1 \\ -S\phi & C\phi & 0 \\ C\phi/C\theta & S\phi/C\theta & 0 \end{bmatrix} \quad [5.65]$$

This matrix is singular when $C\theta = 0$, as already obtained in § 3.6.2.

5.11.4. Quaternions

Differentiating equation [3.34] with respect to time and equating the diagonal elements with those of equation [5.56] leads to the following equation:

$$\begin{cases} 2(Q_1\dot{Q}_1 + Q_2\dot{Q}_2) = (Q_2Q_4 - Q_1Q_3)\omega_y - (Q_2Q_3 + Q_1Q_4)\omega_z \\ 2(Q_1\dot{Q}_1 + Q_3\dot{Q}_3) = (Q_2Q_3 - Q_1Q_4)\omega_z - (Q_3Q_4 + Q_1Q_2)\omega_x \\ 2(Q_1\dot{Q}_1 + Q_4\dot{Q}_4) = (Q_3Q_4 - Q_1Q_2)\omega_x - (Q_2Q_4 + Q_1Q_3)\omega_y \end{cases} \quad [5.66]$$

By differentiating equation [3.31] with respect to time, we obtain:

$$Q_1\dot{Q}_1 + Q_2\dot{Q}_2 + Q_3\dot{Q}_3 + Q_4\dot{Q}_4 = 0 \quad [5.67]$$

From equations [5.66] and [5.67], we deduce that:

$$\dot{\mathbf{X}}_r = \dot{\mathbf{Q}} = [\dot{Q}_1 \ \dot{Q}_2 \ \dot{Q}_3 \ \dot{Q}_4]^T = \mathbf{\Omega}_Q {}^0\boldsymbol{\omega}_n \quad [5.68]$$

with:

$$\mathbf{\Omega}_Q = \frac{1}{2} \begin{bmatrix} -Q_2 & -Q_3 & -Q_4 \\ Q_1 & Q_4 & -Q_3 \\ -Q_4 & Q_1 & Q_2 \\ Q_3 & -Q_2 & Q_1 \end{bmatrix} \quad [5.69]$$

To obtain the inverse relationship, we use the left inverse. While taking into account that $\mathbf{\Omega}_Q^T \mathbf{\Omega}_Q = \frac{1}{4}$, we obtain:

$$\mathbf{\Omega}_Q^+ = 4\mathbf{\Omega}_Q^T \quad [5.70]$$

We note that, since the integration of the angular velocity ${}^0\boldsymbol{\omega}_n$ does not yield an orientation representation, equation [5.69] can be used to obtain $\dot{\mathbf{Q}}$ whose integration gives the orientation by the Quaternion representation.

5.12. Conclusion

In this chapter, we have shown how to obtain the kinematic model of a robot manipulator using the kinematic Jacobian matrix. This model allows us to compute the linear and angular velocities of the end-effector in terms of the joint velocities.

The Jacobian matrix can be decomposed into two or three matrices containing simpler terms.

Then, we have shown how to use the Jacobian matrix to analyze the workspace and the velocity space of a robot. We have also demonstrated how to use the Jacobian matrix to obtain the static model and we have highlighted the duality of this model with the kinematic model. Finally, the kinematic models associated with the various representations of the task coordinates have been established.

The kinematic model can also be used to find a numerical solution to the inverse geometric problem for a general robot. The necessary tool to obtain this solution is the inverse kinematic model, which is the topic of the next chapter.

Chapter 6

Inverse kinematic model of serial robots

6.1. Introduction

The inverse kinematic model gives the joint velocities $\dot{\mathbf{q}}$ for a desired end-effector velocity $\dot{\mathbf{X}}$. This model is equivalent to the inverse differential model, which determines the differential variation of the joint variables $d\mathbf{q}$ corresponding to a given differential displacement of the end-effector coordinates $d\mathbf{X}$. We obtain the inverse kinematic model by solving a system of linear equations analytically or numerically. The analytical solutions, whenever they exist, offer much lower computational complexity than the numerical solutions, but all the singular cases must be considered separately on a case by case basis [Chevallereau 87]. Thus, the computational complexity of numerical methods is compensated by its generality in handling the regular, singular and redundant cases in a unified way.

In this chapter, we present the techniques used to develop an inverse kinematic model for the regular, singular and redundant cases. The analytical solution is developed for the regular case. The numerical methods presented for the other cases are based essentially on the pseudoinverse of the Jacobian matrix. Finally, we show how to take advantage of redundancy in the inverse kinematic problem using a minimum description of tasks. We assume that the reader is familiar with the techniques of solving linear equations, which are exposed in Appendix 4.

6.2. General form of the kinematic model

From equations [5.22] and [5.50], whatever the method used to describe the end-effector coordinates, the direct kinematic model can be expressed as:

$$\dot{\mathbf{X}} = \begin{bmatrix} \mathbf{\Omega}_p & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{\Omega}_r \end{bmatrix} \begin{bmatrix} {}^0\mathbf{R}_i & \mathbf{0}_3 \\ \mathbf{0}_3 & {}^0\mathbf{R}_i \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & -{}^i\hat{\mathbf{L}}_{j,n} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} {}^i\mathbf{J}_{n,j} \dot{\mathbf{q}} \quad [6.1]$$

or in compact form as:

$$\dot{\mathbf{X}} = {}^0\mathbf{J}_x \dot{\mathbf{q}} \quad [6.2]$$

Equation [6.1] can be written as:

$${}^i\dot{\mathbf{X}}_{n,j} = {}^i\mathbf{J}_{n,j} \dot{\mathbf{q}} \quad [6.3]$$

with:

$${}^i\dot{\mathbf{X}}_{n,j} = \begin{bmatrix} \mathbf{I}_3 & {}^i\hat{\mathbf{L}}_{j,n} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} {}^0\mathbf{R}_i & \mathbf{0}_3 \\ \mathbf{0}_3 & {}^0\mathbf{R}_i \end{bmatrix} \begin{bmatrix} \mathbf{\Omega}_p^{-1} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{\Omega}_r^+ \end{bmatrix} \dot{\mathbf{X}} \quad [6.4]$$

We find in § 5.11 the expression of the pseudoinverse $\mathbf{\Omega}_r^+$ for different representations of the orientation, while $\mathbf{\Omega}_p^{-1} = \mathbf{I}_3$ if the Cartesian coordinates are used to describe the position.

Since the elements of ${}^i\mathbf{J}_{n,j}$ are simpler than those of ${}^0\mathbf{J}_x$, equation [6.3] is more appropriate for developing an analytical solution to the inverse kinematic problem. To simplify the notation, we will use the following form for both equations [6.2] and [6.3]:

$$\dot{\mathbf{X}} = \mathbf{J} \dot{\mathbf{q}} \quad [6.5]$$

NOTE.— If $n < 6$, we cannot use the Jacobian matrix ${}^i\mathbf{J}_{n,j}$ systematically. The singularities of this matrix do not take into account the corresponding particular choice of the task coordinates [Borrel 86].

6.3. Inverse kinematic model for a regular case

In this case, the Jacobian matrix \mathbf{J} is square and of full rank. Thus, it is possible to move the end-effector with finite velocity in any desired direction of the task space. The joint velocities can be evaluated using one of the following methods.

6.3.1. First method

We compute \mathbf{J}^{-1} , the inverse of \mathbf{J} , either numerically or analytically. Then, the joint velocity vector $\dot{\mathbf{q}}$ is obtained as:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \dot{\mathbf{X}} \quad [6.6]$$

If the matrix \mathbf{J} has the following form:

$$\mathbf{J} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{C} \end{bmatrix} \quad [6.7]$$

the matrices \mathbf{A} and \mathbf{C} being square and invertible, it is easy to show that:

$$\mathbf{J}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ -\mathbf{C}^{-1}\mathbf{B}\mathbf{A}^{-1} & \mathbf{C}^{-1} \end{bmatrix} \quad [6.8]$$

Consequently, the inverse of \mathbf{J} reduces to the inverse of two matrices of smaller dimension. For a six degree-of-freedom robot with a spherical wrist, the general form of \mathbf{J} is given by equation [6.7] where \mathbf{A} and \mathbf{C} are (3x3) matrices [Gorla 84].

6.3.2. Second method

In this method, instead of solving a linear system of n equations in n unknowns, the problem is reduced to solving two linear systems of equations of lower dimensions. In general, this technique requires less computational complexity. Let us take for example a six degree-of-freedom robot with a spherical wrist whose Jacobian matrix (see Example 5.3) can be written as:

$$\begin{bmatrix} \dot{\mathbf{X}}_a \\ \dot{\mathbf{X}}_b \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0}_3 \\ \mathbf{B} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_b \end{bmatrix} \quad [6.9]$$

\mathbf{A} and \mathbf{C} being (3x3) regular square matrices.

The solution $\dot{\mathbf{q}}$ is given by:

$$\begin{cases} \dot{\mathbf{q}}_a = \mathbf{A}^{-1} \dot{\mathbf{X}}_a \\ \dot{\mathbf{q}}_b = \mathbf{C}^{-1} [\dot{\mathbf{X}}_b - \mathbf{B} \dot{\mathbf{q}}_a] \end{cases} \quad [6.10]$$

which, *a priori*, is simpler than that obtained by the first method.

• **Example 6.1.** Calculate the inverse kinematic model of the Stäubli RX-90 robot. The Jacobian ${}^3\mathbf{J}_6$ has been computed in Example 5.3. We develop the solutions according to equations [6.8] and [6.10].

i) *first method.* The inverses of \mathbf{A} and \mathbf{C} are respectively:

$$\mathbf{A}^{-1} = \begin{bmatrix} 0 & 0 & V1 \\ 0 & V3 & 0 \\ -1/RL4 & V2V3/RL4 & 0 \end{bmatrix}$$

$$\mathbf{C}^{-1} = \begin{bmatrix} V4 & 1 & -V5 \\ S4 & 0 & C4 \\ -C4/S5 & 0 & S4/S5 \end{bmatrix}$$

with:

$$V1 = \frac{1}{S23RL4 - C2D3}$$

$$V2 = -RL4 + S3D3$$

$$V3 = \frac{1}{C3D3}$$

$$V4 = C4 \cotg 5$$

$$V5 = S4 \cotg 5$$

Using equation [6.8], we obtain:

$${}^3\mathbf{J}_6^{-1} = \begin{bmatrix} 0 & 0 & V1 & 0 & 0 & 0 \\ 0 & V3 & 0 & 0 & 0 & 0 \\ -1/RL4 & V2V3/RL4 & 0 & 0 & 0 & 0 \\ -S4C5V7 & V5V6 & V8 & V4 & 1 & -V5 \\ C4/RL4 & -C4V6 & -S23S4V1 & S4 & 0 & C4 \\ S4V7 & -S4V6/S5 & S23C4V1/S5 & -C4/S5 & 0 & S4/S5 \end{bmatrix}$$

with:

$$V6 = \frac{S3}{C3RL4}$$

$$V7 = \frac{1}{S5RL4}$$

$$V8 = (-S23V4 - C23)V1$$

The computation of $\dot{\mathbf{q}}$ by equation [6.8] needs 18 additions, 47 multiplications/divisions and 8 sine/cosine functions;

ii) *second method.* We calculate successively $\dot{\mathbf{q}}_a$ and $\dot{\mathbf{q}}_b$:

$$\dot{\mathbf{q}}_a = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} V1\dot{X}_3 \\ V3\dot{X}_2 \\ (-\dot{X}_1 + V2V3\dot{X}_2) / RL4 \end{bmatrix}$$

$$\dot{\mathbf{X}}_b - \mathbf{B} \dot{\mathbf{q}}_a = \begin{bmatrix} \dot{X}_{4'} \\ \dot{X}_{5'} \\ \dot{X}_{6'} \end{bmatrix} = \begin{bmatrix} \dot{X}_4 - S23 \dot{q}_1 \\ \dot{X}_5 - C23 \dot{q}_1 \\ \dot{X}_6 - \dot{q}_2 - \dot{q}_3 \end{bmatrix}$$

$$\dot{\mathbf{q}}_b = \begin{bmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} = \mathbf{C}^{-1} \begin{bmatrix} \dot{X}_{4'} \\ \dot{X}_{5'} \\ \dot{X}_{6'} \end{bmatrix} = \begin{bmatrix} C4 \cotg5 \dot{X}_{4'} + \dot{X}_{5'} - S4 \cotg5 \dot{X}_{6'} \\ S4 \dot{X}_{4'} + C4 \dot{X}_{6'} \\ (-C4 \dot{X}_{4'} + S4 \dot{X}_{6'}) / S5 \end{bmatrix}$$

This solution requires 12 additions, 22 multiplications/divisions and 8 sine/cosine functions.

6.4. Solution in the neighborhood of singularities

When the robot is non-redundant, the singular configurations are the roots of $\det(\mathbf{J}) = 0$. In the redundant case, they are given by the roots of $\det(\mathbf{J}\mathbf{J}^T) = 0$. Thus, singularities are identified by the rank deficiency of the matrix \mathbf{J} , which physically represents the inability of the robot to generate an arbitrary velocity in the task space. The neighborhood of a singular position is more precisely detected by using the singular values. In fact, the decrease of one or several singular values is generally more significant to indicate the vicinity of a singular configuration than that of examining the value of the determinant. In the neighborhood of these configurations, the use of the classical inverse of the Jacobian matrix will give excessive joint velocities. Since such high velocities are physically unrealizable, we cannot obtain an accurate motion.

The redundancy can be exploited to design robots that avoid singularities [Hollerbach 84b], [Luh 85a]. However, robots with revolute joints will have unavoidable singularities [Baillieul 84]. In § 6.5, we will see that redundancy may be exploited to go away from avoidable singularities [Baillieul 84]. An avoidable singularity is a singular configuration where the corresponding tool location can be reached with a different non-singular configuration.

6.4.1. Use of the pseudoinverse

The most widely proposed methods for solving the inverse kinematic problem near singularities involve the use of the pseudoinverse \mathbf{J}^+ of the matrix \mathbf{J} (Appendix 4):

$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{X}} \quad [6.11]$$

This solution, proposed by Whitney [Whitney 69], minimizes $\|\dot{\mathbf{q}}\|^2$ and $\|\dot{\mathbf{X}} - \mathbf{J} \dot{\mathbf{q}}\|^2$. Depending on $\dot{\mathbf{X}}$, the following cases are distinguished:

- $\dot{\mathbf{X}}$ belongs to $\mathcal{R}(\mathbf{J})$, representing the range space of \mathbf{J} : equation [6.11] gives an exact solution with zero error even though the inverse Jacobian \mathbf{J}^{-1} is not defined;
- $\dot{\mathbf{X}}$ belongs to the subspace of the degenerated directions $\mathcal{R}(\mathbf{J})^\perp$: there are no joint velocities that can generate this velocity. In this case, the solution [6.11] gives $\dot{\mathbf{q}} = \mathbf{0}$.
- $\dot{\mathbf{X}}$ belongs to both $\mathcal{R}(\mathbf{J})$ and $\mathcal{R}(\mathbf{J})^\perp$: the solution [6.11] gives $\dot{\mathbf{q}}$, which only realizes the components belonging to $\mathcal{R}(\mathbf{J})$.

A major shortcoming of this method is that it produces discontinuous joint velocities near singularities [Wampler 86]. This can be seen by expressing the joint velocity solution in terms of singular value decomposition (§ 5.8.1). In fact, far from singularities, the joint velocities are given by:

$$\dot{\mathbf{q}} = \sum_{i=1}^r \frac{1}{s_i} \mathbf{V}_i \mathbf{U}_i^T \dot{\mathbf{X}} \quad [6.12]$$

While approaching a singularity, s_{\min} becomes small, leading to high joint velocities. At singularity, the smallest singular value s_{\min} becomes zero, consequently, it is not taken into account any more. The summation in equation [6.12] is carried out up to $m-1$, and the joint velocity $\dot{\mathbf{q}}$ decreases significantly.

NOTE.— Both $\|\dot{\mathbf{q}}\|^2$ and $\|\dot{\mathbf{X}} - \mathbf{J} \dot{\mathbf{q}}\|^2$ may contain elements with different units. However, using radians for the angles and meters for the distances gives good results for industrial robots of common size (1 to 2 meters reach).

6.4.2. Use of the damped pseudoinverse

A general approach to solving the problem of discontinuity of the pseudoinverse solution at a singular configuration is to use the damped least-squares method, which is known as the Levenberg-Marquardt stabilization method [Wampler 86], [Nakamura 87]. This solution minimizes the following expression:

$$\|\dot{\mathbf{X}} - \mathbf{J}\dot{\mathbf{q}}\|^2 + \alpha^2 \|\dot{\mathbf{q}}\|^2 \quad [6.13]$$

where α is a constant.

This new criterion means that the end-effector tracking error is weighted against the norm of joint velocity by using the factor α , also known as the *damping factor*. This solution is typically obtained as the least-squares solution of the following system:

$$\begin{bmatrix} \mathbf{J} \\ \alpha \mathbf{I}_n \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{X}} \\ \mathbf{0}_{n \times 1} \end{bmatrix} \quad [6.14]$$

The left hand side matrix is of dimension $(m \times n)$ and rank r , using the pseudo inverse formula leads to as:

$$\dot{\mathbf{q}}_d = [\mathbf{J}^T \mathbf{J} + \alpha^2 \mathbf{I}_n]^{-1} \mathbf{J}^T \dot{\mathbf{X}} \quad [6.15]$$

Using svd of \mathbf{J}

$$\dot{\mathbf{q}}_d = [\mathbf{V} \mathbf{S} \mathbf{U}^T \mathbf{U} \mathbf{S}^T \mathbf{V}^T + \alpha^2 \mathbf{I}_n] \mathbf{J}^T \dot{\mathbf{X}} \left[(\mathbf{S}^2 + \alpha^2) \mathbf{I}_n \right] \mathbf{V} \mathbf{S} \mathbf{U}^T \dot{\mathbf{X}} \quad [6.16]$$

After developing, the solution is written as:

$$\dot{\mathbf{q}}_d = \sum_{i=1}^n \frac{s_i}{s_i^2 + \alpha^2} \mathbf{V}_i \mathbf{U}_i^T \dot{\mathbf{X}} \quad [6.17]$$

If $s_i \gg \alpha$, then $\frac{s_i}{s_i^2 + \alpha^2} \approx \frac{1}{s_i}$. If $s_i \ll \alpha$, then $\frac{s_i}{s_i^2 + \alpha^2} \approx \frac{s_i}{\alpha^2}$.

The error due to the damping factor α in the joint coordinates is expressed as:

$$\begin{aligned}
\dot{\mathbf{e}}_q &= \dot{\mathbf{q}} - \dot{\mathbf{q}}_d = \sum_{i=1}^n \left(\frac{1}{s_i} - \frac{s_i}{s_i^2 + \alpha^2} \right) \mathbf{V}_i \mathbf{U}_i^T \dot{\mathbf{X}} \\
&= \sum_{i=1}^n \frac{\alpha^2}{s_i(s_i^2 + \alpha^2)} \mathbf{V}_i \mathbf{U}_i^T \dot{\mathbf{X}}
\end{aligned} \tag{6.18}$$

The error in $\dot{\mathbf{X}}$ is obtained as:

$$\dot{\mathbf{e}}_x = \mathbf{J} \dot{\mathbf{e}}_q = \sum_{i=1}^m \frac{\alpha^2}{s_i^2 + \alpha^2} \mathbf{U}_i \mathbf{U}_i^T \dot{\mathbf{X}} \tag{6.19}$$

The damping factor α limits the norm of the solution. However, at positions far away from singularities, no damping is needed. Thus, a trade-off must be found between the precision of the solution and the possibility of its realization.

Wampler [Wampler 86] proposes to use a fixed damping factor $\alpha = 0.003$, while Nakamura [Nakamura 86] suggests the computation of the damping factor as a function of the manipulability w (equation [5.38]) as follows:

$$\begin{cases} \alpha = \alpha_0 \left(1 - \frac{w}{w_0}\right)^2 & \text{if } w < w_0 \\ \alpha = 0 & \text{if } w \geq w_0 \end{cases} \tag{6.20}$$

where α_0 is a positive constant and w_0 is a threshold, which defines the boundary of the neighborhood of singular points.

A more appropriate solution can be obtained by adjusting the value of α as a function of the smallest singular value s_{\min} , which is the exact measure of the neighborhood of a singular position. Maciejewski and Klein [Maciejewski 88] propose to compute the damping factor as follows:

$$\begin{cases} \alpha = \varepsilon^2 - s_{\min}^2 & \text{if } s_{\min} \leq \varepsilon \\ \alpha = 0 & \text{if } s_{\min} > \varepsilon \end{cases} \tag{6.21}$$

where ε is a constant.

In [Maciejewski 88], we find an efficient method to estimate s_{\min} . In the damping least-squares method, the robot can stay blocked in a singular configuration if the desired velocity is along the degenerated directions, i.e. when (equations [5.30] and [5.31]):

$$\dot{\mathbf{X}} = \sum_{i=r+1}^m \mathbf{U}_i (\mathbf{U}_i^T \dot{\mathbf{X}}) \quad [6.22]$$

where $r < m$ gives the rank of \mathbf{J} .

6.4.3. Other approaches for controlling motion near singularities

The kinematic model, which is a first order linearization, does not give an exact solution respecting the actuator constraints in the neighborhood of singularities. Some authors [Nielsen 91], [Chevallereau 98] have used the IGM or a kinematic model of higher order to determine the joint variables corresponding to a Cartesian motion passing through a singularity. Recently, it has been shown [Lloyd 96] that the end-effector could move along any specified path using a suitable time law.

To show the efficiency of such techniques, let us consider the case of a two degree-of-freedom planar robot in the singular configuration "extended arm". Let us suppose that we want to move the terminal point towards the origin along the x -axis (Figure 6.1a) (which is a degenerated direction for the kinematic model). It is easy to deduce from the kinematic model that a constant velocity motion along this direction is not feasible. However, a motion with a constant end-effector acceleration and a zero initial velocity can be proved realizable (Figure 6.1b) by developing the IGM up to the second order [Nielsen 91] or by using the second-order kinematic model [Chevallereau 98].

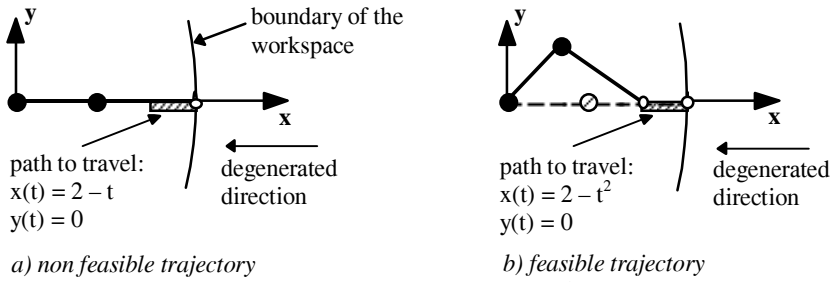


Figure 6.1. Displacement along a degenerated direction

In addition, Egeland and Spangelo [Egeland 91] showed that, in certain cases, a non-feasible path could become realizable after carrying out a specific motion in the null space of \mathbf{J} . This motion does not modify the end-effector coordinates but it modifies the degenerated direction. Let us illustrate this method for the two degree-of-freedom planar robot with identical link lengths. From the initial configuration "folded arm" of Figure 6.2a, it is not possible to track a trajectory along the x

direction. However, after a $\pi/2$ rotation of the first joint, which does not modify the terminal point coordinates but modifies the degenerated direction (Figure 6.2b), we can produce a velocity along the x -axis by using the kinematic model.

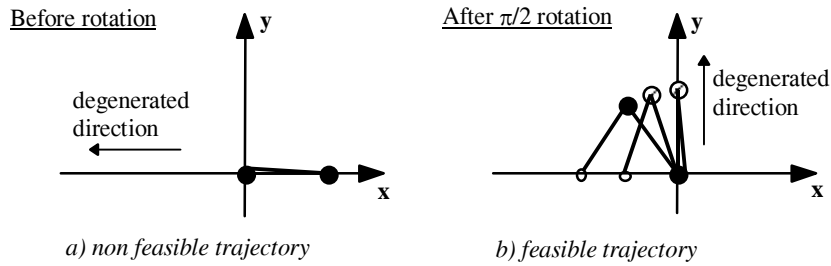


Figure 6.2. Motion in the null space of J

6.5. Inverse kinematic model of redundant robots

A robot manipulator is redundant when its number of degrees of freedom N is greater than the dimension of the workspace M . The difference $(N - M)$ represents the degree of redundancy. In this case, the inverse kinematic model gives an infinite number of solutions. Consequently, secondary performance criteria can be optimized, such as:

- minimizing the norm of the joint velocities [Whitney 69];
- avoiding obstacles [Maciejewski 85], [Baillieul 86];
- avoiding singular configurations [Yoshikawa 84a];
- avoiding joint limits [Fournier 80], [Klein 84];
- minimizing driving joint torques [Baillieul 84], [Hollerbach 85].

When the end-effector coordinates are independent, we have $n = N$ and $m = M$. For a redundant mechanism, the Jacobian J is represented by an $(m \times n)$ matrix, with $n > m$. In the following sections, we present several approaches to solving the inverse kinematic problem of redundant robots.

6.5.1. Extended Jacobian

In this approach, we add $n - m$ secondary linearly independent equations to the end-effector coordinates \mathbf{X} [Baillieul 85], [Chang 86], [Nenchev 92]. These equations can represent either physical constraints on the robot or constraints related to the environment. They are written in the following general form:

$$\mathbf{X}_c = \mathbf{h}(\mathbf{q}) \quad [6.23]$$

In this expression, \mathbf{X}_c is an $((n-m) \times 1)$ vector whose elements are functions of \mathbf{q} . Differentiating equation [6.23] with respect to time gives:

$$\dot{\mathbf{X}}_c = \mathbf{J}_h \dot{\mathbf{q}} \quad [6.24]$$

where $\mathbf{J}_h = \partial \mathbf{h}(\mathbf{q}) / \partial \mathbf{q}$ is the $((n-m) \times n)$ Jacobian matrix of $\mathbf{h}(\mathbf{q})$. Combining this equation with the kinematic model, we obtain an $(n \times n)$ extended Jacobian matrix \mathbf{J}_a and a new velocity vector $\dot{\mathbf{X}}_a$ such that:

$$\dot{\mathbf{X}}_a = \mathbf{J}_a \dot{\mathbf{q}} \quad [6.25]$$

$$\text{with } \dot{\mathbf{X}}_a = \begin{bmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{X}}_c \end{bmatrix} \text{ and } \mathbf{J}_a = \begin{bmatrix} \mathbf{J} \\ \mathbf{J}_h \end{bmatrix}.$$

If the extended Jacobian \mathbf{J}_a is not singular, a unique solution for the joint velocity $\dot{\mathbf{q}}$ is obtained by inverting \mathbf{J}_a . We can use this technique to optimize the desired objective function $\phi(\mathbf{q})$ by taking $\mathbf{h}(\mathbf{q})$ such that:

$$h_i(\mathbf{q}) = 0 = (\eta_i)^T \nabla \phi \quad \text{for } i = 1, \dots, n-m \quad [6.26]$$

where the $(n \times 1)$ vectors η_i , for $i = 1, \dots, n-m$, form a basis for the null space of \mathbf{J} , and $\nabla \phi$ is the gradient of ϕ .

Since the calculation of the basis of the null space of the Jacobian matrix must be carried out analytically, this method can be used only for systems with a small degree of redundancy. A solution to this problem can be found in [Klein 95].

The extended Jacobian method presents the following disadvantages:

- the choice of the $(n-m)$ additional relationships is not a trivial matter;
- the extended Jacobian \mathbf{J}_a may be singular even though the end-effector Jacobian is of full rank. These configurations are called *artificial singularities* or *algorithmic singularities*.

A desirable property of this method is that it yields cyclic behavior, meaning that a closed path in the task space is always tracked by a closed path in the joint space. This is important because it allows one to judge the suitability of a trajectory after executing one cycle.

6.5.2. Use of the pseudoinverse of the Jacobian matrix

The vast majority of research in the control of redundant robots has involved the resolution through the use of the pseudoinverse \mathbf{J}^+ of the Jacobian matrix:

$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{X}} \quad [6.27]$$

This solution minimizes $\|\dot{\mathbf{q}}\|^2$. Because of this minimization property, the early hope of researchers [Whitney 69] was that singularities would automatically be avoided. It has been proved that, without modification, this approach does not avoid singularity [Baillieul 85]. Moreover, Klein and Huang [Klein 83] have pointed out that it does not produce cyclic behavior, which is a serious practical problem.

For these reasons, we generally add to the pseudoinverse solution another component belonging to the null space of the Jacobian, in order to realize the secondary objective function.

6.5.3. Weighted pseudoinverse

Since each joint has different limits and even different units, it may be interesting to weight the contribution of each joint in the objective function differently. This can be achieved by the use of the weighted pseudoinverse, which minimizes a criteria C such that:

$$C = \dot{\mathbf{q}}^T \mathbf{E} \dot{\mathbf{q}} \quad [6.28]$$

When \mathbf{J} is of full rank, the solution is given by:

$$\dot{\mathbf{q}} = \mathbf{J}_E^+ \dot{\mathbf{X}} \quad [6.29]$$

with:

$$\mathbf{J}_E^+ = \mathbf{E}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{E}^{-1} \mathbf{J}^T)^{-1} \quad [6.30]$$

Benoit et al. [Benoit 75] propose to take for \mathbf{E} the inertia matrix of the robot (Chapter 9) in order to minimize the kinetic energy. Konstantinov et al. [Konstantinov 81] have used the weighted pseudoinverse to avoid the joint limits.

6.5.4. Pseudoinverse solution with an optimization term

One of the advantages of the pseudoinverse solution is the possibility to utilize the null space to optimize another objective function (beside that of $\|\dot{\mathbf{q}}\|^2$). In fact, the general solution of the linear system [6.5] is written as (Appendix 4):

$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{X}} + (\mathbf{I}_n - \mathbf{J}^+ \mathbf{J}) \mathbf{Z} \quad [6.31]$$

where \mathbf{Z} is an arbitrary (nx1) vector in the $\dot{\mathbf{q}}$ space.

The second term on the right belongs to the null space of \mathbf{J} . It corresponds to a self-motion of the joints that does not move the end-effector. This term, which is called *homogeneous solution* or *optimization term*, can be used to optimize a desired function $\phi(\mathbf{q})$. In fact, taking $\mathbf{Z} = \alpha \nabla \phi$ where $\nabla \phi$ is the gradient of this function with respect to \mathbf{q} , minimizes the function $\phi(\mathbf{q})$ when $\alpha < 0$ and maximizes it when $\alpha > 0$. Equation [6.31] is rewritten as:

$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{X}} + \alpha (\mathbf{I}_n - \mathbf{J}^+ \mathbf{J}) \nabla \phi \quad [6.32]$$

with:

$$\nabla \phi = \left[\frac{\partial \phi}{\partial q_1} \quad \dots \quad \frac{\partial \phi}{\partial q_n} \right]^T \quad [6.33]$$

The value of α allows us to realize a trade-off between the minimization of $\|\dot{\mathbf{q}}\|^2$ and the optimization of $\phi(\mathbf{q})$. In the following sections, we present two examples of desired objective functions.

6.5.4.1. Avoiding joint limits

A practical solution to control a redundant robot is to keep the joint variables away from their limits \mathbf{q}_{\max} and \mathbf{q}_{\min} . Let:

$$\mathbf{q}_{\text{moy}} = \frac{1}{2} (\mathbf{q}_{\max} + \mathbf{q}_{\min}) \quad [6.34]$$

where \mathbf{q}_{moy} is the mean value of the joint positions, and:

$$\Delta \mathbf{q} = \mathbf{q}_{\max} - \mathbf{q}_{\min} \quad [6.35]$$

A possible scalar function, whose minimization generates a motion away from the joint limits, can be expressed in the following quadratic form [Fournier 80]:

$$\phi(\mathbf{q}) = \sum_{i=1}^n \left[\frac{q_i - q_{i_{\text{moy}}}}{\Delta q_i} \right]^2 \quad [6.36]$$

The division by Δq_i allows us to weight the contribution of each joint in $\phi(\mathbf{q})$ such that it varies between 0 and 1. The i^{th} element of the vector \mathbf{Z} is written as (with $\alpha < 0$):

$$Z_i = \frac{\alpha \partial \phi(\mathbf{q})}{\partial q_i} = \frac{2\alpha(q_i - q_{i_{\text{moy}}})}{\Delta q_i^2} \quad [6.37]$$

NOTE.— If the mean position of a joint corresponds to a singular configuration, it is recommended to replace the corresponding value of $q_{i_{\text{moy}}}$ by another value.

About the criterion [6.36], Klein [Klein 84] pointed out that the quadratic form, used generally to solve optimization problems, does not always give the best solution to the desired objectives. To avoid joint limits in particular, the following form is more suitable:

$$\phi = \max \frac{|q_i - q_{i_{\text{moy}}}|}{|\Delta q_i|} \quad \text{for } i = 1, \dots, n \quad [6.38]$$

Introducing this criterion in equation [6.32] is however not as easy as the quadratic criterion. A solution consists of approximating the criterion [6.38] by a p-norm function defined as [Klein 83]:

$$\|\mathbf{q} - \mathbf{q}_{\text{moy}}\|_p = \left[\sum_{i=1}^n |q_i - q_{i_{\text{moy}}}|^p \right]^{1/p} \quad [6.39]$$

When p tends towards infinity, the corresponding p-norm meets the criterion [6.38]. However, sufficient approximation can be achieved by taking $p = 6$.

6.5.4.2. Increasing the manipulability

In § 5.8.2, we showed that the manipulability $w(\mathbf{q})$ of a robot manipulator (equation [5.38]) could be used as a measure of the ability of the mechanism to move its end-effector. At a singular point, w is minimum and is zero. In order to

improve the manipulability of a structure, we can choose to maximize a scalar function ϕ such that:

$$\phi(\mathbf{q}) = \det [\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q})] \quad [6.40]$$

We calculate \mathbf{Z} as indicated previously with $\alpha > 0$. Maximizing ϕ moves the robot away from the singular configurations.

NOTE.- Certain singular configurations are unavoidable [Baillieul 84]. This is the case if there is no other configuration that can yield the same end-effector location. For the three degree-of-freedom planar robot of Example 6.1, the unavoidable singularities correspond to the configurations where it is fully stretched out or folded up (Figure 6.3). The other singularities are avoidable and the robot can find other configurations to achieve them (Figure 6.4).

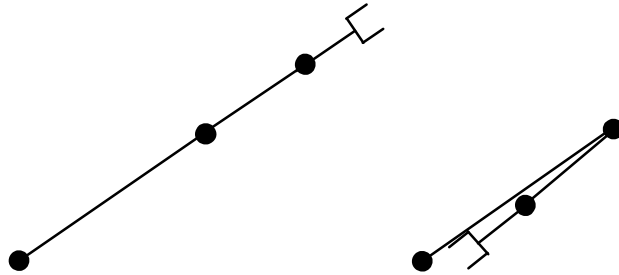


Figure 6.3. Unavoidable singularities of a three degree-of-freedom planar robot

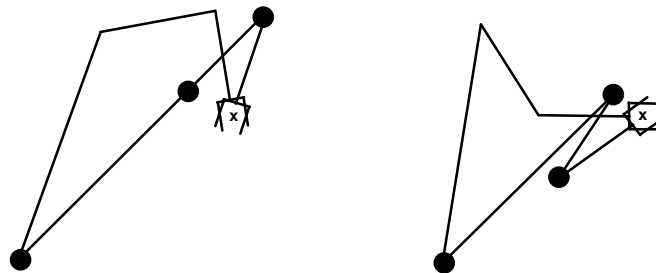


Figure 6.4. Avoidable singularities of a three degree-of-freedom planar robot

6.5.5. Task-priority concept

To solve the inverse kinematic model of redundant robots, Nakamura [Nakamura 87] introduced the concept of task priority, where a required task is

divided into a primary task \mathbf{X}_1 of higher priority and a secondary task \mathbf{X}_2 of lower priority. These tasks are described by the following relationships:

$$\mathbf{X}_1 = \mathbf{f}_1(\mathbf{q}) \quad [6.41]$$

$$\mathbf{X}_2 = \mathbf{f}_2(\mathbf{q}) \quad [6.42]$$

Let m_1 and m_2 be the dimensions of \mathbf{X}_1 and \mathbf{X}_2 respectively. Differentiating equations [6.41] and [6.42] with respect to time gives:

$$\dot{\mathbf{X}}_1 = \mathbf{J}_1 \dot{\mathbf{q}} \quad [6.43]$$

$$\dot{\mathbf{X}}_2 = \mathbf{J}_2 \dot{\mathbf{q}} \quad [6.44]$$

where $\mathbf{J}_i = \partial \mathbf{f}_i(\mathbf{q}) / \partial \mathbf{q}$ is the $(m_i \times n)$ Jacobian matrix of the task \mathbf{X}_i . Using the pseudoinverse, the general solution of the two tasks can be obtained by :

$$\dot{\mathbf{q}} = \mathbf{J}_1^+ \dot{\mathbf{X}}_1 + (\mathbf{I}_n - \mathbf{J}_1^+ \mathbf{J}_1) \mathbf{J}_2^+ \dot{\mathbf{X}}_2$$

In this equation the first term realizes the first task, whereas the second term is the projection of the second task on the null space of the first task. This method is not applicable to more than two tasks. In the following we present a recursive solution for a task which is composed of any number of subtasks Nakamura [], Mansard[].

Iteration number 1 is the solution of the first task using the pseudo of equation [6.43], it is given by:

$$\dot{\mathbf{q}}^1 = \mathbf{J}_1^+ \dot{\mathbf{X}}_1$$

where the exponent of $\dot{\mathbf{q}}$ denotes the iteration number.

The second task will be obtained through a joint velocity \mathbf{Z}_1 projected on the null space of the first task, such that:

$$\dot{\mathbf{q}}^2 = \mathbf{J}_1^+ \dot{\mathbf{X}}_1 + (\mathbf{I}_n - \mathbf{J}_1^+ \mathbf{J}_1) \mathbf{Z}_1 \quad [6.45]$$

Substituting equation [6.45] into equation [6.44], we obtain \mathbf{Z}_1 in terms of $\dot{\mathbf{X}}_2$:

$$\mathbf{J}_2 (\mathbf{I}_n - \mathbf{J}_1^+ \mathbf{J}_1) \mathbf{Z}_1 + \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{X}}_1 = \dot{\mathbf{X}}_2 \quad [6.46]$$

From this equation, the vector \mathbf{Z}_1 can be determined in terms of $\dot{\mathbf{X}}_2$ by using the pseudoinverse:

$$\mathbf{Z}_1 = (\mathbf{J}_2 \mathbf{P}_1)^+ [\dot{\mathbf{X}}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{X}}_1] \quad [6.47]$$

where $\mathbf{P}_1 = (\mathbf{I}_n - \mathbf{J}_1^+ \mathbf{J}_1)$, and $(\mathbf{J}_2 \mathbf{P}_1)$ is an $(m_2 \times n)$ matrix.

The joint velocity $\dot{\mathbf{q}}$ of the robot is obtained from equations [6.45] and [6.47]:

$$\dot{\mathbf{q}}^2 = \mathbf{J}_1^+ \dot{\mathbf{X}}_1 + \mathbf{P}_1 (\mathbf{J}_2 \mathbf{P}_1)^+ [\dot{\mathbf{X}}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{X}}_1] \quad [6.48]$$

Since $\mathbf{P} \mathbf{P} = \mathbf{P}$ and $\mathbf{P}^T = \mathbf{P}$, thus $\mathbf{P}^+ = \mathbf{P}$, $\mathbf{P} \mathbf{P}^+ = \mathbf{P}^+$ the previous relation can be simplified as:

$$\dot{\mathbf{q}}^2 = \mathbf{J}_1^+ \dot{\mathbf{X}}_1 + (\mathbf{J}_2 \mathbf{P}_1)^+ [\dot{\mathbf{X}}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{X}}_1]$$

$$\dot{\mathbf{q}}^2 = \mathbf{J}_1^+ \dot{\mathbf{X}}_1 + (\mathbf{J}_2 \mathbf{P}_1)^+ [\dot{\mathbf{X}}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{X}}_1] \quad [6.48]$$

In case of a third task, the corresponding \mathbf{Z}_2 , will be obtained in terms of $\dot{\mathbf{X}}_3$ by solving the equation of the third task:

$$\dot{\mathbf{q}}^3 = \mathbf{J}_1^+ \dot{\mathbf{X}}_1 + (\mathbf{J}_2 \mathbf{P}_1)^+ [\dot{\mathbf{X}}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{X}}_1] + \mathbf{P}_2 \mathbf{Z}_2 = \dot{\mathbf{q}}^2 + \mathbf{P}_2 \mathbf{Z}_2$$

Giving

$$\mathbf{J}_3 \dot{\mathbf{q}}^3 = \mathbf{J}_3 (\dot{\mathbf{q}}^2 + \mathbf{P}_2 \mathbf{Z}_2) = \dot{\mathbf{X}}_3$$

Thus

$$\mathbf{Z}_2 = (\mathbf{J}_3 \mathbf{P}_2)^+ [\dot{\mathbf{X}}_3 - \mathbf{J}_3 \dot{\mathbf{q}}^2]$$

$$\dot{\mathbf{q}}^3 = \dot{\mathbf{q}}^2 + \mathbf{P}_2 (\mathbf{J}_3 \mathbf{P}_2)^+ [\dot{\mathbf{X}}_3 - \mathbf{J}_3 \dot{\mathbf{q}}^2] = \dot{\mathbf{q}}^2 + (\mathbf{J}_3 \mathbf{P}_2)^+ [\dot{\mathbf{X}}_3 - \mathbf{J}_3 \dot{\mathbf{q}}^2]$$

This recursive relation can be generalized for arbitrary number of tasks.

The interpretation of this method is illustrated in Figure 6.5. At each iteration i the next task is realized thanks to \mathbf{Z}_{i-1} which is projected on the null space of the augmented matrix of all the previous tasks.

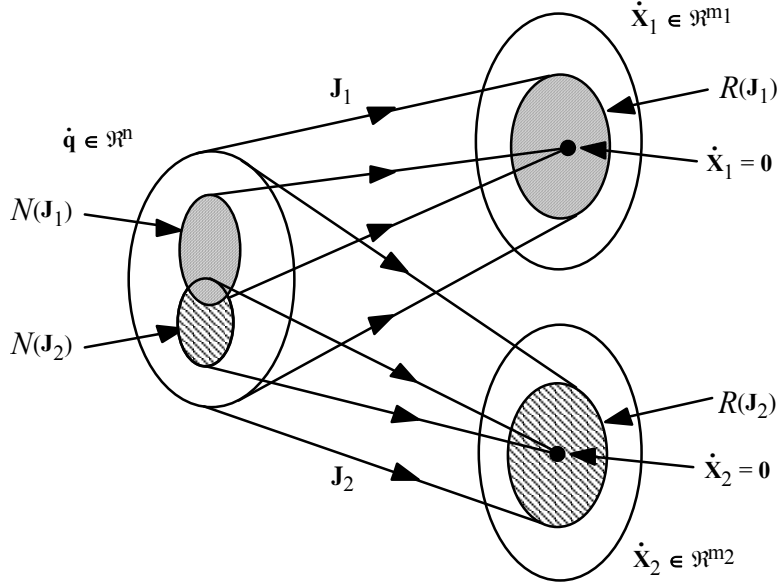


Figure 6.5. Null space and range space of tasks X_1 and X_2 (from [Nakamura 87])

6.6. Numerical calculation of the inverse geometric problem

When it is not possible to find a closed-form solution to the inverse geometric problem, we can use the differential model to compute an iterative numerical solution. Two numerical methods can be used to obtain the joint positions \mathbf{q}^d corresponding to a desired location ${}^0\mathbf{T}_n^d$ of the terminal link, we proceed as follows:

First method: This problem can be solved using a nonlinear optimization algorithm to find \mathbf{q} minimizing criterion: $C = \| {}^0\mathbf{T}_n^d - {}^0\mathbf{T}_n(\mathbf{q}) \|$, a solution is obtained if the corresponding $C = 0$.

Second method: The optimization problem is solved using iterative method using the inverse differential method as follows:

- initialize \mathbf{q}^c by the current joint configuration or by any random value within the joint domain of the robot;
- calculate the location of the terminal frame ${}^0\mathbf{T}_n^c$ corresponding to \mathbf{q}^c using the direct geometric model;
- calculate the vectors of position error \mathbf{dX}_p and rotation error \mathbf{dX}_r , representing the difference between the desired location ${}^0\mathbf{T}_n^d$ and the current location ${}^0\mathbf{T}_n^c$.

Note that $\mathbf{dX}_p = \mathbf{dP}_n = \mathbf{P}_n^d - \mathbf{P}_n^c$ and $\mathbf{dX}_r = \mathbf{u} \alpha$, where the angle α and the unit

vector \mathbf{u} are obtained by solving the equation (§ 2.3.8): ${}^0\mathbf{R}_n^d = \mathbf{rot}(\mathbf{u}, \alpha) {}^0\mathbf{R}_n^c$, which can be written as ${}^0\mathbf{R}_n^d ({}^0\mathbf{R}_n^c)^T = \mathbf{rot}(\mathbf{u}, \alpha)$;

- if the absolute values of the elements of $({}^0\mathbf{T}_n^d - {}^0\mathbf{T}_n^c)$ are sufficiently small, then $\mathbf{q}^d = \mathbf{q}^c$ and stop the calculation;
- to remain in the validity domain of the differential model, which is a first order expansion, we must introduce thresholds S_p and S_r on $d\mathbf{X}_p$ and $d\mathbf{X}_r$ respectively such that:

– if $\|d\mathbf{X}_p\| > S_p$, then $d\mathbf{X}_p = \frac{d\mathbf{X}_p}{\|d\mathbf{X}_p\|} S_p$

– if $\|d\mathbf{X}_r\| > S_r$, then $d\mathbf{X}_r = \frac{d\mathbf{X}_r}{\|d\mathbf{X}_r\|} S_r$

The values 0.2 meter and 0.2 radian for these thresholds are acceptable for most of the industrial robots in view of their dimensions;

- calculate the Jacobian matrix ${}^0\mathbf{J}_n(\mathbf{q}^c)$ denoted as \mathbf{J} ;
- calculate the joint variation $d\mathbf{q} = \mathbf{J}^+ d\mathbf{X}$. An optimization term in the null space of \mathbf{J} can also be taken into account;
- update the current joint configuration: $\mathbf{q}^c = \mathbf{q}^c + d\mathbf{q}$;
- return to the second step.

This algorithm converges rapidly and can be executed in real time. If it does not converge within a relatively large number of iterations, or to obtain another solution, we have to restart the calculation using a new random value \mathbf{q}^c ; if no convergence occurs for many different values of \mathbf{q}^c , it can be stated that there is no solution.

6.7. Minimum description of tasks [Fournier 80], [Dombre 81]

In current robot controllers, the desired trajectory of the end-effector is described by a sequence of frames. However, in many industrial applications, it is not necessary to completely specify the location of the end-effector frame and the task could be described by a reduced number of coordinates. For example:

- when the manipulated object is symmetric: for a spherical object, it is not necessary to specify the orientation; likewise, the rotation of a cylindrical object about its axis can be left free;
- releasing an object into a container: if the end-effector is already above the container, only an approach distance has to be specified; the task is thus described by a translational component;

- transferring objects from one point to another with arbitrary orientation; the task can be described by three translational components;
- placing a cylindrical object on a conveyor: the only orientation constraint is that the principal axis of the cylinder is horizontal; if the end-effector is already above the conveyor, the task could be described by two components (one vertical translation and one rotation).

When the number of components of a task is less than the number of degrees of freedom of the robot, the robot is redundant with respect to the task. Consequently, an infinite number of solutions can be obtained to realize such tasks. This redundancy can be exploited to satisfy secondary optimization criteria (§ 6.5).

6.7.1. Principle of the description

The proposed description of task is minimal in the sense that it only constrains the degrees of freedom of the task that have a functional role. The formulation is based on the use of the contact conditions between usual surfaces (plane, cylinder, sphere) that describe usual mechanical joints (or pairing) (Table 6.1 and Figure 6.6). To these six joints, we add the composite revolute and prismatic joints, which have one degree of mobility (Figure 6.7), and the fixed rigid pairing, which has no degree of freedom.

The description of a task is realized by a sequence of virtual mechanical joints. The choice of a type of joint is dictated by the local constraints associated with the task.

Table 6.1. Simple mechanical joints

	Plane	Cylinder	Sphere
Plane	Plane contact	Line contact	Point contact
Cylinder		Cylindrical joint	Cylindrical groove joint
Sphere			Spherical joint

A practical description of the mechanical joint formulation consists of specifying the task in terms of contact between two simple geometric entities (point, line, plane), one belonging to the robot, the other to the environment [Dombre 85]. A spherical joint, for example, is specified by matching two points. In the same way, the revolute and prismatic joints will be specified with two simultaneous combinations of geometric elements. The choice is not unique: a revolute joint for

example can be achieved either by a line-to-line contact and a point-to-plane contact simultaneously or by a line-to-line contact and a point-to-point contact.

This geometric description is particularly convenient for graphic programming of tasks. Figure 6.8 shows the example of a peg-in-hole assembly, realized with the CAD/CAM software package CATIA [Catia] in which this formulation was implemented for robotic application. The different steps are as follows:

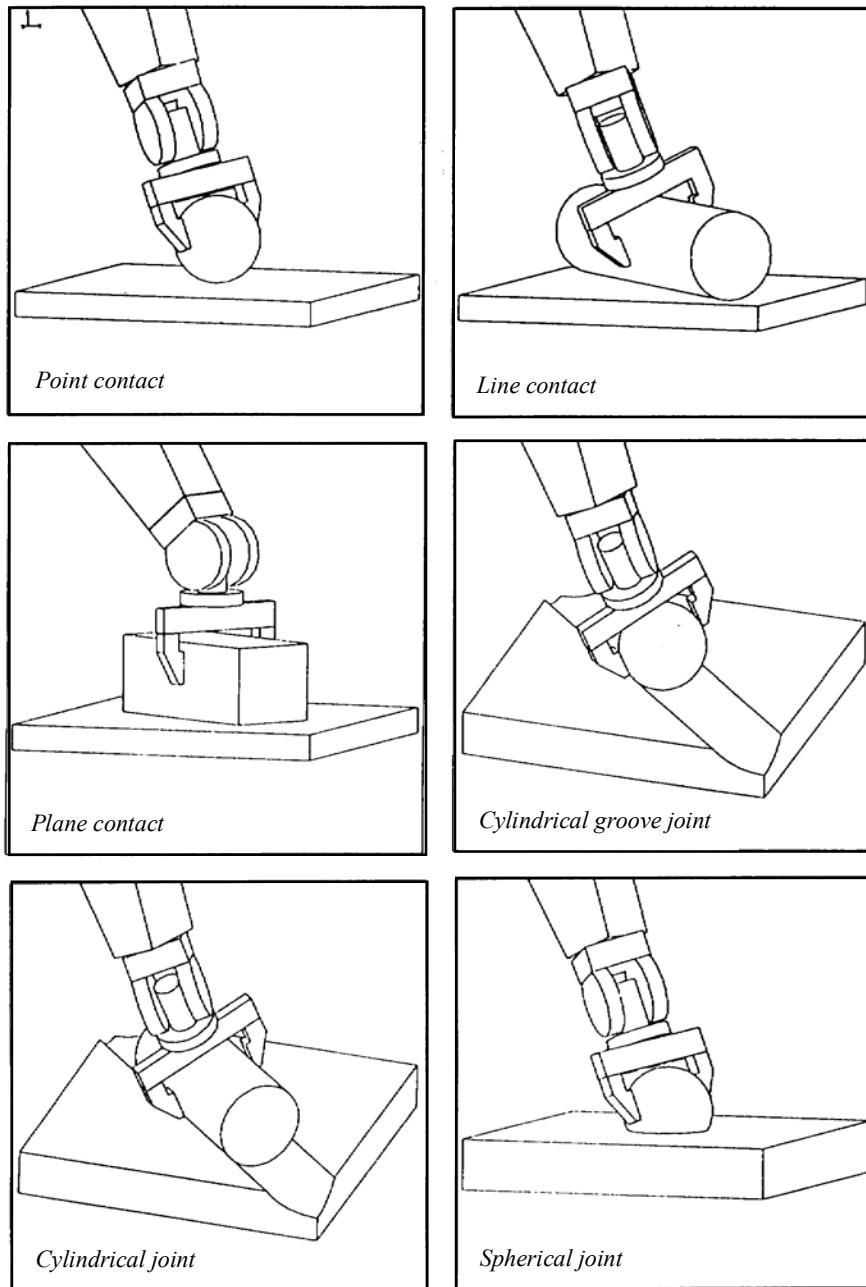


Figure 6.6. Simple mechanical joints

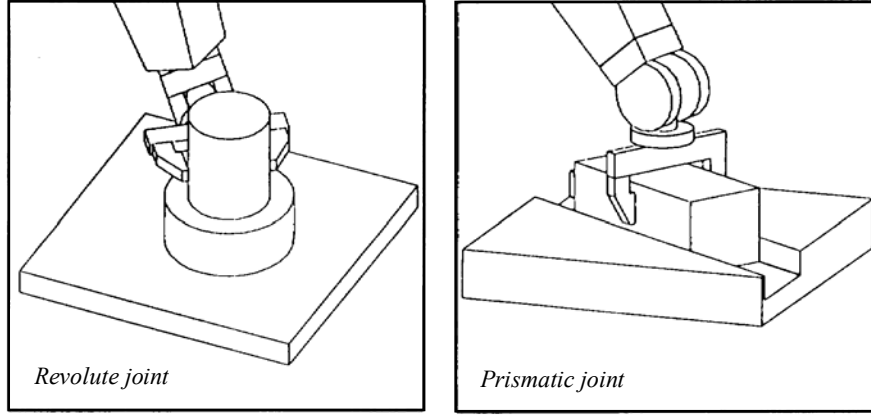


Figure 6.7. *Revolute and prismatic joints*

- 1) definition of a point-to-point contact (spherical joint) by selecting a point of the robot and a point of the environment (Figure 6.8a); after execution, the cylinder is positioned with an arbitrary orientation above the assembly site (Figure 6.8b);
- 2) definition of a line-to-line contact (cylindrical contact) by selecting a line of the robot and a line of the environment (Figure 6.8b); after execution, the axes of the hole and the peg are aligned (Figure 6.8c);
- 3) definition of a revolute joint by selecting a point and a line of the robot, and a point and a line of the environment (Figure 6.8c); after execution, the assembly task is completed (Figure 6.8d).

6.7.2. *Differential models associated with the minimum description of tasks*

To implement these types of tasks, we write the differential model of the location of frame R_E in the following form:

$$\begin{aligned}
 \begin{bmatrix} {}^0d\mathbf{P}_E \\ {}^0\delta_E \end{bmatrix} &= \begin{bmatrix} {}^0\mathbf{R}_n & \mathbf{0}_3 \\ \mathbf{0}_3 & {}^0\mathbf{R}_n \end{bmatrix} \begin{bmatrix} {}^n d\mathbf{P}_E \\ {}^n \delta_E \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{R}_n & \mathbf{0}_3 \\ \mathbf{0}_3 & {}^0\mathbf{R}_n \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & -{}^n\hat{\mathbf{P}}_E \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} {}^n d\mathbf{P}_n \\ {}^n \delta_n \end{bmatrix} \\
 &= \begin{bmatrix} {}^0\mathbf{R}_n & -{}^0\mathbf{R}_n {}^n\hat{\mathbf{P}}_E \\ \mathbf{0}_3 & {}^0\mathbf{R}_n \end{bmatrix} {}^n \mathbf{J}_n d\mathbf{q}
 \end{aligned} \tag{6.49}$$

where ${}^n\mathbf{P}_E$ defines the origin of frame R_E referred to frame R_n .

The differential model of a virtual joint can be written as:

$$d\mathbf{X} = \mathbf{H} {}^n\mathbf{J}_n d\mathbf{q} \quad [6.50]$$

where ${}^n\mathbf{J}_n$ and \mathbf{H} are $(6 \times n)$ and $(c \times 6)$ matrices respectively, and c indicates the number of constraint equations of the task.

We will show in the following section how to determine \mathbf{H} for the virtual joints [Dombre 81].

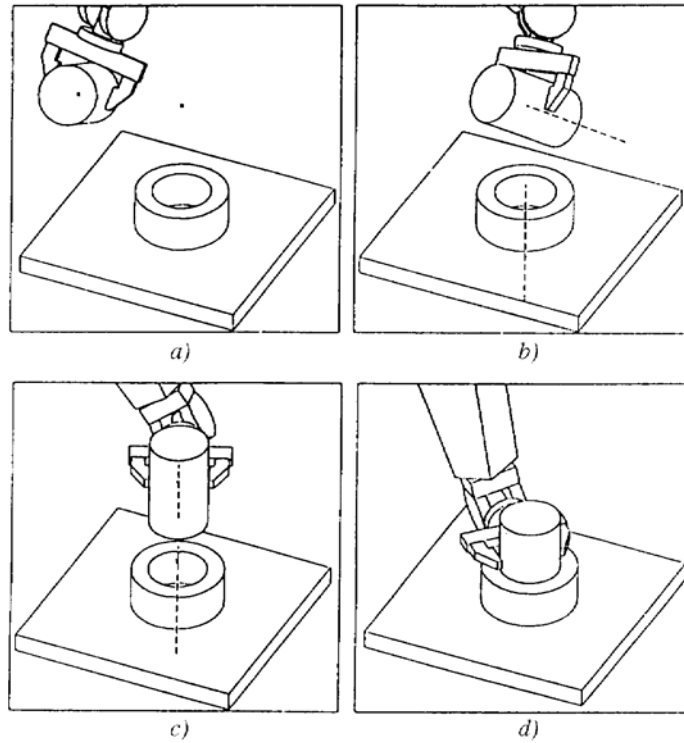


Figure 6.8. Graphic programming of an assembly task with a minimum description

6.7.2.1. Point contact (point on plane)

This joint drives a point O_E of the tool on any position on a plane Q (Figure 6.9). Let \mathbf{N} be the unit vector normal to the plane Q and let O_D be an arbitrary point of Q . The necessary global displacement to realize the point contact is expressed in frame R_0 by:

$$\mathbf{r} = {}^0\mathbf{N}^T [{}^0\mathbf{P}_D - {}^0\mathbf{P}_E] \quad [6.51]$$

where ${}^0\mathbf{P}_D$ and ${}^0\mathbf{P}_E$ define the coordinates of the points O_D and O_E in frame R_0 .

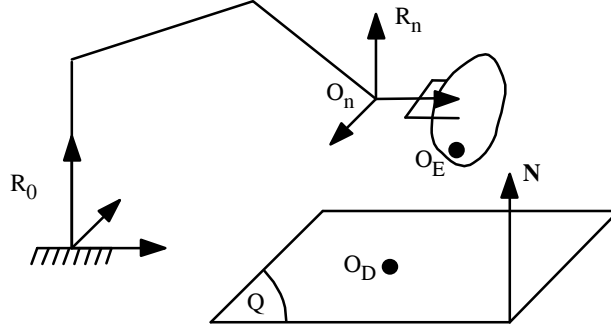


Figure 6.9. Realization of point contact

The displacement \mathbf{r} is realized by a sequence of elementary displacements along a single direction such that (equation [6.49]):

$$\begin{aligned} d\mathbf{X} = d\mathbf{r} &= {}^0\mathbf{N}^T d\mathbf{P}_E = \begin{bmatrix} {}^0\mathbf{N}^T {}^0\mathbf{R}_n & -{}^0\mathbf{N}^T {}^0\mathbf{R}_n {}^n\hat{\mathbf{P}}_E \end{bmatrix} {}^n\mathbf{J}_n d\mathbf{q} \\ &= {}^0\mathbf{N}^T {}^0\mathbf{R}_n \begin{bmatrix} \mathbf{I}_3 & -{}^n\hat{\mathbf{P}}_E \end{bmatrix} {}^n\mathbf{J}_n d\mathbf{q} \end{aligned} \quad [6.52]$$

Expression [6.52] constitutes the differential model of the point contact. The matrix \mathbf{H} is given by the row vector ${}^0\mathbf{N}^T {}^0\mathbf{R}_n \begin{bmatrix} \mathbf{I}_3 & -{}^n\hat{\mathbf{P}}_E \end{bmatrix}$.

6.7.2.2. Line contact (line on plane)

The equations of a line contact are derived from Figure 6.10. The line U_E is driven on plane Q without constraining its orientation in the plane. We can realize this joint by simultaneously carrying out a rotation and a translation [Dombre 88a]. However, it is more judicious to avoid the calculation of an angle by defining the task as driving two points O_{E1} and O_{E2} of U_E on plane Q . The joint is thus equivalent to two point contact. The corresponding differential model is written as:

$$d\mathbf{X} = \begin{bmatrix} dr_1 \\ dr_2 \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{N}^T {}^0\mathbf{R}_n & -{}^0\mathbf{N}^T {}^0\mathbf{R}_n {}^n\hat{\mathbf{P}}_{E1} \\ {}^0\mathbf{N}^T {}^0\mathbf{R}_n & -{}^0\mathbf{N}^T {}^0\mathbf{R}_n {}^n\hat{\mathbf{P}}_{E2} \end{bmatrix} {}^n\mathbf{J}_n d\mathbf{q} \quad [6.53]$$

where \mathbf{H} is a $(2 \times n)$ matrix.

We can generalize this approach for the other joints where the j^{th} row of \mathbf{H} takes the following general form:

$$\mathbf{H}_j = {}^0\mathbf{N}_j^T {}^0\mathbf{R}_n \begin{bmatrix} \mathbf{I}_3 & -{}^n\hat{\mathbf{P}}_{Ej} \end{bmatrix} \quad [6.54]$$

where ${}^0\mathbf{N}_j$ denotes the unit vector of the normal to the plane of the j^{th} point contact, and ${}^n\mathbf{P}_{Ej}$ is the vector of the coordinates of the tool point O_{Ej} with respect to frame R_n .

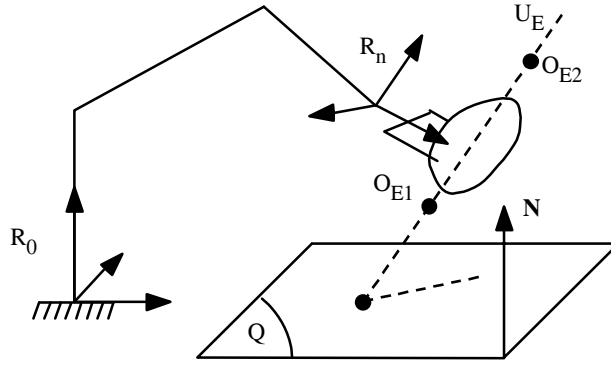


Figure 6.10. Realization of line contact

6.7.2.3. Planar contact (plane on plane)

This joint drives a plane Q_E attached to the tool on a plane Q_D of the environment, without orientation or position constraints (Figure 6.11). We select three non-aligned points O_{E1} , O_{E2} and O_{E3} in Q_E , then we carry out three simultaneous point contacts.

6.7.2.4. Cylindrical groove joint (point on line)

The cylindrical groove joint drives a point O_E of the tool on a line U_D of the environment. This is done by simultaneously realizing two point contacts of O_E on two arbitrary orthogonal planes Q_{D1} and Q_{D2} whose intersection is the line U_D (Figure 6.12).

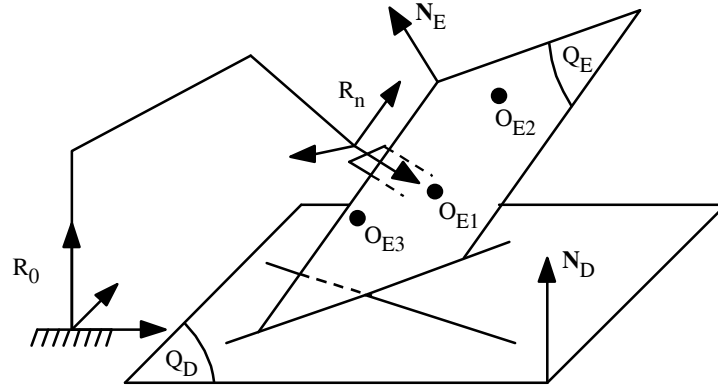


Figure 6.11. Realization of a plane contact

6.7.2.5. Cylindrical joint (line on line)

The task consists of aligning two lines U_E and U_D without position or orientation constraints along and about these lines (Figure 6.12). We define two arbitrary orthogonal planes Q_{D1} and Q_{D2} whose intersection is the line U_D and whose normals are N_{D1} and N_{D2} respectively. To realize a cylindrical joint, any two distinct points O_{E1} and O_{E2} of the line U_E are driven simultaneously on the planes Q_{D1} and Q_{D2} . In other words, the cylindrical joint corresponds to four point contacts.

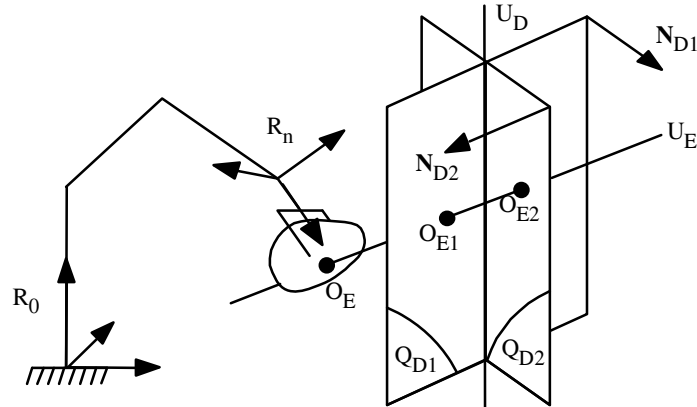


Figure 6.12. Realization of cylindrical groove joint, cylindrical joint and revolute joint

6.7.2.6. Spherical joint (point on point)

A spherical joint drives a point O_E of the tool on a point O_D of the environment without constraining the orientation of the tool. The task can be realized by three point contacts that drive O_E simultaneously on the planes Q_{D1} , Q_{D2} and Q_{D3} , which are parallel to the planes (y_0, z_0) , (x_0, z_0) , (x_0, y_0) and pass through the point O_D . The required displacements r_1 , r_2 and r_3 are the components of the vector $\mathbf{O}_E\mathbf{O}_D$ along the axes of frame R_0 . The task is defined as:

$$\mathbf{dX} = \begin{bmatrix} dr_1 \\ dr_2 \\ dr_3 \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{R}_n & -{}^0\mathbf{R}_n {}^n\hat{\mathbf{p}}_E \end{bmatrix} {}^n\mathbf{J}_n \mathbf{dq} \quad [6.55]$$

6.7.2.7. Revolute joint (line-point on line-point)

A revolute joint (Figure 6.12) consists of aligning a line U_E of the tool with a line U_D of the environment while simultaneously driving a point O_E of U_E on a plane Q_D normal to U_D (not represented in the figure). Let O_{E1} and O_{E2} be any two distinct points on U_E . Similar to the cylindrical groove joint, let us consider that Q_{D1} and Q_{D2} are two arbitrary orthogonal planes whose intersection is the line U_D . The joint is thus equivalent to the simultaneous realization of five point contacts:

- driving the point O_{E1} on the planes Q_{D1} and Q_{D2} ;
- driving the point O_{E2} on the planes Q_{D1} and Q_{D2} ;
- driving the point O_E on the plane Q_D .

In practice, it is more convenient to describe the revolute joint by a line-to-line contact and a point-to-point contact. This choice leads to seven equations, and the rank of the matrix \mathbf{HJ} is five.

6.7.2.8. Prismatic joint (plane-plane on plane-plane)

A prismatic joint consists of aligning two lines of the tool with two geometrically compatible lines of the environment, and making a translation along an arbitrary axis. To simplify, we consider that the two lines are perpendicular and the displacement is carried out along one of these lines.

Let U_{E1} and U_{E2} be the two lines of the tool and let U_{D1} and U_{D2} be two compatible lines of the environment (Figure 6.13). Let us suppose that the free translation is along the line U_{D1} . Let Q_{D1a} and Q_{D1b} be two arbitrary orthogonal

planes whose intersection is U_{D1} , and O_{E1a} and O_{E1b} be any two distinct points on the line U_{E1} . We realize the prismatic joint by five point contacts:

- driving the point O_{E1a} on the planes Q_{D1a} and Q_{D2b} ;
- driving the point O_{E1b} on the planes Q_{D1a} and Q_{D2b} ;
- driving any point of U_{E2} , that is not the intersection of U_{E1} and U_{E2} , on the plane formed by the lines U_{D1} and U_{D2} .

Similar to the revolute joint case, it may be more convenient for the user to specify a prismatic joint using two plane-to-plane contacts. In this case, the number of equations is six.

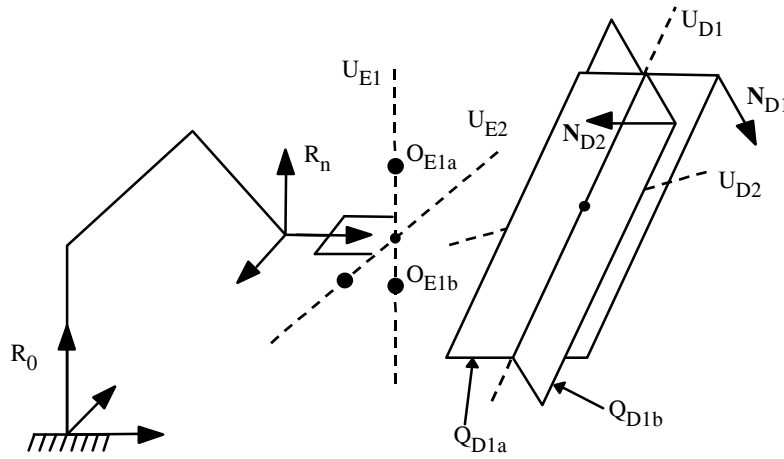


Figure 6.13. *Realization of a prismatic joint*

NOTES.—

- for the fixed rigid pairing, we use the complete description of $d\mathbf{X} = \mathbf{J} d\mathbf{q}$;
- Table 6.2 summarizes the specification of each virtual mechanical joint as well as the number of necessary equations.

Table 6.2. *Equivalence between virtual mechanical joints and geometric specification*

Type of joint	Elements of the tool	Elements of the environment	Number of independent equations	Total number of equations
Point contact	Point	Plane	1	1
Line contact	Line	Plane	2	2
Plane contact	Plane	Plane	3	3
Cylindrical groove	Point	Line	2	2
Spherical	Point	Point	3	3
Cylindrical	Line	Line	4	4
Revolute	Line-Point	Line-Point	5	7
Prismatic	Plane-Plane	Plane-Plane	5	6

6.8. Conclusion

In this chapter, we have studied the inverse kinematic model by considering the regular, singular and redundant cases. The solution may be obtained either analytically or numerically. The analytical solution can be used for simple robots in regular configurations, whereas the numerical methods are more general.

We have also shown how to reduce the functional degrees of freedom of the task using a description method based on the virtual mechanical joints formulation.

The redundancy, whether it is a built-in feature of the robot or the consequence of a minimum description of the task, can be used to optimize the trajectory generation of the mechanism. In this respect, the solution based on the pseudoinverse method proves to be very powerful. It allows us to realize secondary optimization functions such as keeping the joints away from their limits or improving the manipulability.

References

- [[Ait Ahmed 93](#)] Ait-Ahmed, M., Contribution à la modélisation géométrique et dynamique des robots parallèles. Ph.D. thesis, 1993, LAAS, Toulouse.
- [**Ait Mohamed 95**] Ait Mohamed A., "Commande dynamique de robots redondants dans l'espace opérationnel", Thèse de Doctorat, Université de Nantes et Ecole Centrale de Nantes, France, February 1995.
- [**Aldon 82**] Aldon M.J., "Elaboration automatique de modèles dynamiques de robots en vue de leur conception et de leur commande", Thèse d'Etat, USTL, Montpellier, France, October 1982.
- [**Aldon 86**] Aldon M.J., "Identification des paramètres structuraux des robots manipulateurs", *Proc. Outils Mathématiques pour la Modélisation et la Commande des Robots*, Paris, September 1986, p. 243-296.
- [**An 85**] An C.H., Atkeson C.G., Hollerbach J.M., "Estimation of inertial parameters of rigid body links of manipulators", *Proc. 24th IEEE Conf. on Decision and Control*, Fort Lauderdale, December 1985, p. 990-995.
- [**An 87**] An C.H., Hollerbach J.M., "Kinematic stability issues in force control of manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, March-April 1987, p. 897-903.
- [**Angeles 88**] Angeles J., Gosselin C., "Détermination du degré de liberté des chaînes cinématiques", *Trans. of the CSME*, Vol. 12, 1977, p. 219-226.
- [[Angeles 2002](#)] Angeles J., *Fundamentals of Robotic Mechanical Systems*, Springer-Verlag, New York, 2002.
- [**Arimoto 84**] Arimoto S., Miyazaki F., "Stability and robustness of PID feedback control for robots manipulators of sensory capability", *The 1st Int. Symp. of Robotics Research*, MIT Press, Cambridge, 1984.
- [**Arimoto 93**] Arimoto S., Liu Y.H., Naniwa T., "Model-based adaptive hybrid control for geometrically constrained robots", *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, May 1993, p. 618-623.
- [**Armstrong 79**] Armstrong W.W., "Recursive solution to the equation of motion of an N-links manipulator", *Proc. 5th World Congress on Theory of Machines and Mechanisms*, Montréal, 1979, p. 1343-1346.

- [**Armstrong 86**] Armstrong B., Khatib O., Burdick J., "The explicit dynamic model and inertial parameters of the PUMA 560 Arm", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 510-518.
- [**Armstrong 88**] Armstrong B., "Dynamics for robot control: friction modeling and ensuring excitation during parameter identification", Ph. D Thesis, Dept. of Electrical Engineering, Stanford University, May 1988.
- [**Armstrong 89**] Armstrong B., "On finding exciting trajectories for identification experiments involving systems with non linear dynamics", *The Int. J. of Robotics Research*, Vol. 8(6), 1989, p. 28-48.
- [**Armstrong 91**] Armstrong B., *Control of Machines with frictions*, Kluwer Academic Publishers, 1991.
- [**Armstrong 94**] Armstrong-Hélouvry B., Dupont P., Canudas de Wit C., "A survey of analysis tools and compensation methods for the control of machines with friction", *Automatica*, Vol. 30(10), 1994, p. 1083-1138.
- [**Asada 86**] Asada H., Slotine J.-J.E., *Robot analysis and control*, John Wiley & Sons, New York, 1986.
- [**Atkeson 86**] Atkeson C.G., An C.H., Hollerbach J.M., "Estimation of inertial parameters of manipulator loads and links", *The Int. J. of Robotics Research*, Vol. 5(3), 1986, p. 101-119.
- [**Aubin 91**] Aubin A., "Modélisation, identification et commande du bras manipulateur TAM", Thèse de Doctorat, INPG, Grenoble, France, 1991.
- [**Baillieul 84**] Baillieul J., Hollerbach J.M., Brockett R., "Programming and control of kinematically redundant manipulators", *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, December 1984, p. 768-774.
- [**Baillieul 85**] Baillieul J., "Kinematic programming alternatives for redundant manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, St Louis, March 1985, p. 722-728.
- [**Baillieul 86**] Baillieul J., "Avoiding obstacles and resolving kinematic redundancy", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 1698-1704.
- [**Baron 94**] Baron L., Angeles J., "The decoupling of the direct kinematics of parallel manipulators using redundant sensors", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Diego, May 1994, p. 974-979.
- [**Bartels 88**] Bartels R.H., Beaty J.C., Barsky B.A., *Mathématiques et CAO 6, B-splines*, Hermès, Paris, 1988.
- [**Baumgarte 72**] Baumgarte J., "Stabilization of constraints and integral of motion", *Computer Methods in Applied Mech. Eng.*, Vol. 1(1), 1972, p. 1-16.
- [**Bayard 88**] Bayard D., Wen J.T., "New class of control laws for robotic manipulators; Part 2: Adaptive case", *Int. J. Control*, Vol. 47(5), 1988, p. 1387-1406.
- [**Bejczy 74**] Bejczy A.K., "Robot arm dynamics and control", NASA Technical Memorandum 33-669, Jet Propulsion Laboratory, Pasadena, 1974.
- [**Bejczy 79**] Bejczy A.K., "Dynamic models and control equations for manipulators", Tutorial Workshop, 18th IEEE Conf. on Decision and Control, Fort Lauderdale, December 1979.
- [**Bejczy 85**] Bejczy A.K., Tarn T.J., Yun X., Hans S., "Non linear feedback control of Puma 560 robot arm by computer", *Proc. 24th IEEE Conf. on Decision and Control*, Fort Lauderdale, December 1985, p. 1680-1688.

- [Beji 97] Beji L., "Modélisation, identification et commande d'un robot parallèle", Thèse de Doctorat, Université d'Evry-Val d'Essone, France, June 1997.
- [Benallegue 91] Benallegue A., "Contribution à la commande dynamique adaptative des robots manipulateurs rapides", Thèse de Doctorat, Université Pierre et Marie Curie, Paris, France, November 1991.
- [Benhlila 93] Benhlila, S., "Identification des paramètres dynamiques des systèmes mécaniques articulés complexes", Thèse de Doctorat, ENSAM, Paris, France, 1993.
- [Bennett 03] Bennett G.T., "A new mechanism", *Engineering*, Vol. 76, 1903., p. 777-778.
- [Bennis 91a] Bennis F., "Contribution à la modélisation géométrique et dynamique des robots à chaîne simple et complexe", Thèse de doctorat, E.N.S.M, Nantes, France, 1991.
- [Bennis 91b] Bennis F., Khalil W., "Minimum inertial parameters of robots with parallelogram closed-loops", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-21(2), 1991, p. 318-326.
- [Bennis 93] Bennis F., Khalil W., "Modèle géométrique inverse des robots à chaîne découplable : application aux équations de contraintes des boucles fermées", *Trans. of the Canadian Society for Mechanical Engineering*, Vol. 17(4A), 1993, p. 473-491.
- [Benoit 75] Benoit M., Briot M., Donnarel H., Liégeois A., Meyer M.A., Renaud M., "Synthèse de la commande dynamique d'un téléopérateur redondant", *Revue RAIRO*, Série J-2, May 1975, p. 89-103.
- [Berghuis 93] Berghuis H., "Model-based robot control: from theory to practice", Ph. D. Thesis, University of Twente, Enschede, Holland, 1993.
- [Bhattacharya 97] Bhattacharya S., Hatwal H., and Ghosh A.: An on-line estimation scheme for generalized Stewart platform type parallel manipulators, *J. Mechanism and Machine Theory* **32**(1), 1997, p. 79-89.
- [Bhattacharya 98] Bhattacharya S., Nenchev D.N., and Uchiyama M.: A recursive formula for the inverse of the inertia matrix of a parallel manipulator, *J. Mechanism and Machine Theory* **33**(7) , 1998, p. 957-964.
- [Bernhardt 93] Bernhardt R., Albright S.L., *Robot calibration*, Chapman & Hall, London, 1993.
- [Besnard 99] Besnard S., Khalil W., "Calibration of parallel robots using two inclinometers", *Proc. IEEE Int. Conf. on Robotics and Automation*, Detroit, May 1999, p. 1758-1763.
- [Besnard 00] Besnard S., "Etalonnage géométrique des robots série et parallèles", Thèse de Doctorat, Université de Nantes, France, 1996.
- [Besnard 01] Besnard S., Khalil W., "Identifiable parameters for parallel robots kinematic calibration", *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, May 2001, p. ???
- [Bicchi 98] Bicchi A., "Manipulability of cooperating robots with passive joints", *Proc. IEEE Int. Conf. on Robotics and Automation*, Louvain, Belgium, May 1998, p. 1038-1044.
- [Binford 77] Binford T.O. et al., "Discussion of trajectory calculation methods", in 'Exploratory study of computer integrated assembly system', Stanford Artificial Intelligence Lab., Progress Report, Memo AIM-285.4, Stanford, June 1977.
- [Blanchon 87] Blanchon J.-L., "Génération et identification des lois de mouvement en robotique : application à l'identification et à la correction de trajectoires du manipulateur PUMA 560, Thèse de Doctorat, USTL, Montpellier, France, March 1987.

- [**Borm 91**] Borm J.H., Menq C.H., "Determination of optimal measurement configurations for robot calibration based on observability measure", *The Int. J. of Robotics Research*, Vol. 10(1), p. 51-63, 1991.
- [**Borrel 79**] Borrel P., "Modèle de comportement de manipulateurs ; application à l'analyse de leurs performances et à leur commande automatique", Thèse de Troisième Cycle, USTL, Montpellier, France, December 1979.
- [**Borrel 86**] Borrel P., "Contribution à la modélisation géométrique des robots-manipulateurs ; application à la conception assistée par ordinateur", Thèse d'Etat, USTL, Montpellier, France, July 1986.
- [**Boullion 71**] Boullion T.L., Odell P.L., *Generalized inverse matrices*, John Wiley & Sons, New York, 1971.
- [**Bouzouia 89**] Bouzouia B., "Commande des robots manipulateurs : identification des paramètres et étude des stratégies adaptatives", Thèse de Doctorat, UPS, Toulouse, France, May 1989.
- [**Boyer 94**] Boyer F., "Contribution à la modélisation et à la commande dynamique des robots flexible", Thèse de Doctorat, Université Pierre et Marie Curie, Paris, France, 1994.
- [**Boyer 98**] Boyer F., Khalil W., "An efficient calculation of flexible manipulator inverse dynamics", *The Int. J. of Robotics Research*, Vol. 17(3), 1998, p. 282-293.
- [**Boyer 2006**] Boyer F., Porez M., Khalil W., "Macro-continuous Computed Torque Algorithm for a three-dimensional Eel-Like Robot", *IEEE Transaction, Robotics*, Vol: 22 (4), August 2006, p. 763- 775.
- [**Brandl 86**] Brandl H., Johanni R., Otter M., "A very efficient algorithm for the simulation of robots and multibody systems without inversion of the mass matrix", *Proc. IFAC Symp. on Theory of Robots*, Vienna, Austria, December 1986, p. 365-370.
- [**Brogliato 91**] Brogliato B., "Systèmes passifs et commande adaptative des manipulateurs", Thèse de Doctorat, INPG, Grenoble, France, 1991.
- [**Bruyninckx 98**] Bruyninckx H., "Closed-form forward position kinematics for a $(3-1-1-1)^2$ fully parallel manipulator", *IEEE Trans. on Robotics and Automation*, Vol. RA-14(2), 1998, p. 326-328.
- [**Burdick 86**] Burdick J.W., "An algorithm for generation of efficient manipulator dynamic equations", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 212-218.
- [**Burdick 88**] Burdick J.W., "Kinematic analysis and design of redundant manipulators", Ph. D Thesis, Stanford University, 1988.
- [**Caenen 93**] Caenen J.-L., "Contribution à l'identification de paramètres géométriques et non géométriques d'un modèle de robot. Application à l'amélioration de la précision de positionnement statique", Thèse de Doctorat, Université de Valenciennes et de Hainault-Cambrésis, France, January 1993.
- [**Cannon 84**] Cannon H., Schmitz E., "Initial experiments on the end-point control of a flexible one-link robot", *The Int. J. of Robotics Research*, Vol. 3(3), 1984, p. 62-75.
- [**Canudas de Wit 89**] Canudas de Wit C., Noël P., Aubin A., Brogliato B., Drevet P., "Adaptive Friction compensation in robot manipulators: low-velocities", *Proc. IEEE Int. Conf. on Robotics and Automation*, Scottsdale, May 1989, p. 1352-1357.

- [**Canudas de Wit 90**] Canudas de Wit C., Seront V., "Robust adaptive friction compensation", *Proc. IEEE Int. Conf. on Robotics and Automation*, Cincinnati, May 1990, p. 1383-1389.
- [**Canudas de Wit 92**] Canudas de Wit C., Fixot N., Aström K.J., "Trajectory tracking in robot manipulators via nonlinear estimated feedback", *IEEE Trans. on Robotics and Automation*, Vol. RA-8(1), 1992, p. 138-144.
- [**Castain 84**] Castain R.H., Paul R.P., "An on-line dynamic trajectory generator", *The Int. J. of Robotics Research*, Vol. 3(1), 1984, p. 68-72.
- [**Carricato 2009**] Carricato M., Gosselin C., "On the modeling of leg constraints in the dynamic analysis of Gough/Stewart-type platforms", *ASME Journal of Computational and Nonlinear Dynamics*, Vol.4, No.1, 2009
- [**Castelain 86**] Castelain J.M., "Application de la méthode hypercomplexe aux modélisations géométriques et différentielles des robots constitués d'une chaîne cinématique simple", Thèse d'Etat, Université de Valenciennes et du Hainaut-Cambresis, France, December 1986.
- [**Catia**] Dassault Systèmes, 308 Bureaux de la Colline, 92210 Saint Cloud, France.
- [**Cesareo 84**] Cesareo G., Nicolo F., Nicosia S., "DYMIR: a code for generating dynamic model of robots", *Proc. IEEE Int. Conf. on Robotics*, Atlanta, March 1984, p. 115-120.
- [**Chace 67**] Chace M.A., "Analysis of the time dependance of multi-freedom mechanical system in relative coordinate", *Trans. of ASME, J. of Engineering for Industry*, Vol. 89, February 1967, p. 119-125.
- [**Chace 71**] Chace M.A., Bayazitoglu Y.O., "Development and application of a generalized d'Alembert force for multi-freedom mechanical system", *Trans. ASME, J. of Engineering for Industry*, Vol. 93, February 1971, p. 317-327.
- [**Chand 85**] Chand S., Doty K.L., "On-line polynomial trajectories for robot manipulators", *The Int. J. of Robotics Research*, Vol. 4(2), 1985, p. 38-48.
- [**Chang 86**] Chang P.H., "A closed form solution for the control of manipulators with kinematic redundancy", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 9-14.
- [**Charentus 90**] Charentus S., "Modélisation et commande d'un robot manipulateur redondant composé de plusieurs plates-formes de Stewart", Thèse de Doctorat, UPS, Toulouse, France, April 1990.
- [**Chebychev 1854**] Chebychev, P. A. "Théorie des mécanismes connus sous le nom de parallélogrammes, 1ère partie." Mémoires présentées à l'Académie imperial des sciences de Saint-Petersbourg par divers savants, 1854.
- [**Chedmail 86**] Chedmail P., Gautier M., Khalil W., "Automatic modelling of robots including parameters of links and actuators", *Proc. IFAC Symp. on Theory of Robots*, Vienna, Austria, December 1986, p. 295-299.
- [**Chedmail 90a**] Chedmail P., "Contribution à la conception des robots et à la modélisation et commande de robots souples", Thèse d'Etat, ENSM, Nantes, France, January 1990.
- [**Chedmail 90b**] Chedmail P., Gautier M., "Optimum choice of robot actuators", *Trans. of ASME, J. of Engineering for Industry*, Vol. 112(4), 1990, p. 361-367.

- [**Chedmail 92**] Chedmail P., Dombre E., "CAO et robotique : conception et programmation des cellules robotisées", *Revue d'Automatique et de Productique Appliquées*, Vol. 5(2), 1992, p. 27-38.
- [**Chedmail 98**] Chedmail P., Dombre E., Wenger P., *La CAO en robotique : outils et méthodologies*, Collection 'Etudes en Mécanique des Matériaux et des Structures', Hermès, Paris, 1998.
- [**Cheok 93**] Cheok K.C., Overholt J.L., Beck R.R., "Exact methods for determining the kinematics of a Stewart platform using additional displacement sensors", *J. of Robotic Systems*, Vol. 10(5), 1993, p. 974-979.
- [**Cherki 96**] Cherki B., "Commande des robots manipulateurs par retour d'état estimé", Thèse de Doctorat, Université de Nantes et Ecole Centrale de Nantes, France, 1996.
- [**Chevallereau 87**] Chevallereau C., Khalil W., "Efficient method for the calculation of the pseudo inverse kinematic problem", *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, March-April 1987, p. 1842-1848.
- [**Chevallereau 88**] Chevallereau C., "Contribution à la commande des robots-manipulateurs dans l'espace opérationnel", Thèse de Doctorat, ENSM, Nantes, France, May 1988.
- [**Chevallereau 98**] Chevallereau C., "Feasible trajectories in task space from a singularity for a non redundant or redundant robot manipulator", *The Int. J. of Robotic Research*, Vol. 17(1), 1998, p. 56-69.
- [**Chiaverini 93**] Chiaverini S., Sciavicco L., "The parallel approach to force/position control of robotic manipulators", *IEEE Trans. on Robotics and Automation*, Vol. RA-9(4), 1993, p. 361-373.
- [**Clavel 89**] Clavel R., "Une nouvelle structure de manipulateur parallèle pour la robotique légère", *Revue APII-AFCET*, Vol. 23, 1989, p. 501-519.
- [**Cloutier 95**] Cloutier B.P., Pai D.K., Ascher U.M., "The formulation stiffness of forward dynamics algorithms and implications for robot simulation", *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, May 1995, p. 2816-2822.
- [**Codourey 97**] Codourey A., and Burdet E.: A body oriented method for finding a linear form of the dynamic equations of fully parallel robot, IEEE Conf. on Robotics and Automation, 1997, Albuquerque, New Mexico, U.S, p.1612-1619.
- [**Coiffet 81**] Coiffet P., *Les Robots ; Tome 1 : Modélisation et commande*, Hermès, Paris, 1981.
- [**Colbaugh 93**] Colbaugh R., Seraji H., Glass K., "Direct adaptive impedance control of robot manipulators", *J. of Robotic Systems*, Vol. 10, 1993, p. 217-248.
- [**Corke 96**] P. Corke, "In situ measurement of robot motor electrical constants," *Robotica*, vol. 23, no. 14, pp.433-436, 1996
- [**Craig 86a**] Craig J.J., *Introduction to robotics: mechanics and control*, Addison Wesley Publishing Company, Reading, 1986.
- [**Craig 86b**] Craig J.J., Hsu P., Sastry S., "Adaptive control of mechanical manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 190-195.
- [**Craig 93**] Craig J.J., "Calibration of industrial robots", *Proc. 24th Int. Symp. on Industrial Robots*, Tokyo, November 1993, p. 889-893.

- [**Dafaoui 94**] Dafaoui E., "Modélisation et commande d'un robot parallèle : application au suivi de contour", Thèse de Doctorat, Paris XII-Val de Marne, France, September 1994.
- [**Dafaoui 98**] Dafaoui El-M., Amirat Y., Pontnau J., and François C.: Analysis and Design of a Six-DOF Parallel Manipulator, Modeling, Singular Configurations, and Workspace, *IEEE Trans. on Robotics and Automation* **14**(1), 1998, p.78-92.
- [**Dahl 77**] Dahl P.R., "Measurements of solid friction parameters of ball bearings", *Proc. of the 6th Annual Symp. on Incremental Motion Control Systems and Devices*, University of Illinois, 1977.
- [**Damak 96**] Damak M., "Théorie et instrumentation pour l'étalonnage statique des robots : vers une programmation hors-ligne industriellement plus efficace", Thèse de Doctorat, ENSAM, Lille, France, July 1996.
- [**Dasgupta 98**] Dasgupta B., and Mruthyunjaya T.S., "Closed-form dynamic equations of the general Stewart platform through the Newton-Euler approach", *J. Mechanism and Machine Theory* **33**(7), p. 993-1012, October 1998.
- [**Dasgupta 98**] Dasgupta B., and Mruthyunjaya T.S., "A Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator", *J. Mechanism and Machine Theory* **33**(8), P. 1135-1152, 1998.
- [**Dasgupta 99**] Dasgupta B., and Choudhury P., "A general strategy based on the Newton-Euler approach for the dynamic formulation of parallel manipulators", *J. Mechanism and Machine Theory* **34**(6), p. 801-824, 1999.
- [**Daney 00**] Daney D., "Etalonnage géométrique des robots parallèles", Thèse de Doctorat, Université de Nice-Sophia Antipolis, France, February 2000.
- [**de Boor 78**] de Boor C., *A practical guide to splines*, Springer-Verlag, New York, 1978.
- [**de Casteljau 87**] de Casteljau P., *Les quaternions*, Hermès, Paris, 1987.
- [**Dégoulange 93**] Dégoulange E., "Commande en effort d'un robot manipulateur à deux bras : application au contrôle de la déformation d'une chaîne cinématique fermée", Thèse de Doctorat, Université Montpellier II, France, December 1993.
- [**de Larminat 77**] de Larminat P., Thomas Y., *Automatique des systèmes linéaires ; Tome 2 : Identification*, Editions Flammarion, 1977.
- [**Delignières 87**] Delignières S., "Choix de morphologies de robot", Thèse de Docteur-Ingénieur, ENSM, Nantes, France, November 1987.
- [**de Luca 91a**] de Luca A., Oriolo G., "Issues in acceleration resolution of robot redundancy", *Proc. IFAC Symp. on Robot Control, SYROCO'91*, Vienna, Austria, 1991, p. 665-670.
- [**de Luca 91b**] de Luca A., Manes C., "Hybrid force-position control for robots in contact with dynamic environments", *Proc. IFAC Symp. on Robot Control, SYROCO'91*, Vienna, Austria, 1991, p. 177-182.
- [**Denavit 55**] Denavit J., Hartenberg R.S., "A kinematic notation for lower pair mechanism based on matrices", *Trans. of ASME, J. of Applied Mechanics*, Vol. 22, June 1955, p. 215-221.
- [**Desbats 90**] Desbats P., "Modélisation et commande dynamique des robots rapides", Thèse de Doctorat, Université de Paris Sud, Orsay, France, June 1990.

- [**de Schutter 88**] de Schutter J., van Brussel H., "Compliant robot motion - II - a control approach based on external control loop", *The Int. J. of Robotics Research*, Vol. 7(4), 1988, p. 18-33.
- [**Desoer 75**] Desoer C.A., Vidyaasagar M., *Feedback systems: input-output properties*, Academic Press, 1975.
- [**Dietmaier 98**] Deitmaier P., "The Stewart-Gough platform of general geometry can have 40 real postures", in *Advances in Robot Kinematics: Analysis and Control*, J. Lenarcic, M.L. Husty Eds, Klumer Academic Publishers, 1998, p. 7-16.
- [**Dillon 73**] Dillon S.R., "Computer assisted equation generation in linkage dynamics", Ph. D. Thesis, Ohio State University, August 1973.
- [**Dombre 81**] Dombre E., "Analyse des performances des robots-manipulateurs flexibles et redondants ; contribution à leur modélisation et à leur commande", Thèse d'Etat, USTL, Montpellier, France, June 1981.
- [**Dombre 85**] Dombre E., Borrel P., Liégeois A., "A CAD system for programming and simulating robot's actions", in *Computing Techniques for Robots*, I. Aleksander Ed., Kogan Page, London, 1985, p. 222-247.
- [**Dombre 88a**] Dombre E., Khalil W., *Modélisation et commande des robots*, Hermès, Paris, 1988.
- [**Dombre 88b**] Dombre E., "Outils d'aide à la conception de cellules robotisées", in *Techniques de la robotique : perception et planification*, Hermès, Paris, 1988, p. 255-291.
- [**Dombre 94**] Dombre E., Fournier A., "Yet another calibration technique to reduce the gap between CAD world and real world", *Proc. 1st WAC'94 on Intelligent Automation and Soft Computing*, Hawaï, USA, August 1994, p. 47-52.
- [**Dombre 2007**] [Dombre E., Khalil W., *Robot manipulators: Modeling, Performance Analysis and Control*, ISTE,UK, London, Jan. 2007.](#)
- [**Dongarra 79**] Dongarra J.J., Moler C.B., Bunch J.R., Stewart G.W., "LINPACK User's Guide", Philadelphia, SIAM, 1979.
- [**Douss 96**] Douss M., "Programmation hors ligne par CAO-Robotique : caractérisation de lois de contrôleurs de robots et étalonnage de cellules robotisées en vue d'améliorer la précision", Thèse de Doctorat, Université de Franche-Comté, Besançon, France, November 1996.
- [**Drake 77**] Drake S.H., "Using compliance in lieu of sensory feedback for automatic assembly", Ph. D. Thesis, Dept. of Mechanical Engineering, MIT, September 1977.
- [**Driels 90**] Driels M. R., Pathre U.S., "Significance of observation strategy on the design of robot calibration experiment", *J. of Robotic Systems*, Vol. 7, 1990, p. 197-223.
- [**Dubowsky 79**] Dubowsky S., Des Forges D.T., "The application of model-referenced adaptive control to robotic manipulators", *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 101, 1979, p. 193-200.
- [**Duffy 90**] Duffy J., "The fallacy of modern hybrid control theory that is based on 'orthogonal complements' of twist and wrench spaces", *J. of Robotic Systems*, Vol. 7, 1990, p. 139-144.
- [**Edwall 82**] Edwall C.W., Pottinger H.J., Ho C.Y., "Trajectory generation and control of a robot arm using spline functions", *Proc. Robot-6*, Detroit, 1982, p. 421-444.

- [**Egeland 91**] Egeland O., Spangelo I., "Manipulator control in singular configurations-Motion in degenerate directions", in *Advanced Robot Control, Lecture Notes in Control and Information Sciences*, Springer-Verlag, New York, 1991, p. 296-306.
- [**El Omri 96**] El Omri J., "Analyse géométrique et cinématique des mécanismes de type manipulateur", Thèse de Doctorat, Université de Nantes et Ecole Centrale de Nantes, France, February 1996.
- [**El Serafi 91a**] El Serafi K., Khalil W., "Energy based indirect adaptive control of robots", *Proc. IEEE Int. Conf. on Robotics and Automation*, Sacramento, April 1991, p. 2142-2147.
- [**El Serafi 91b**] El Serafi K., "Contribution à la commande adaptative des robots manipulateurs", Thèse de Doctorat, ENSM, Nantes, France, May 1991.
- [**Everett 88**] Everett L.J., Suryoadiprjo A.H., "A study of kinematic models for forward calibration of manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, Philadelphia, April 1988, p. 798-800.
- [**Everett 89**] Everett L.J., "Forward calibration of closed loop jointed manipulators", *The Int. J. of Robotics Research*, Vol. 8(4), August 1989, p. 85-91.
- [**Eykhoff 74**] Eykhoff P., *System identification: parameter and state estimation*, John Wiley & Sons, London, 1974.
- [**Fages 98**] Fages G., "Statistiques 97", *RobAut*, n° 21, March-April 1998, p. 28-32.
- [**Featherstone 83a**] Featherstone R., "Position and velocity transformations between robot end-effector coordinates and joint angles", *The Int. J. of Robotics Research*, Vol. 2(2), 1983, p. 35-45.
- [**Featherstone 83b**] Featherstone R., "The calculation of robot dynamics using articulated-body inertias", *The Int. J. of Robotics Research*, Vol. 2(3), 1983, p. 87-101.
- [**Ferreira 84**] Ferreira E.P., "Contribution à l'identification de paramètres et à la commande des robots manipulateurs", Thèse de Docteur-Ingénieur, UPS, Toulouse, France, July 1984.
- [**Fichter 86**] Fichter E.F., "A Stewart platform-based manipulator: general theory and practical construction", *The Int. J. of Robotic Research*, Vol. 5(2), 1986, p. 157-181.
- [**Fisher 92**] Fisher W., Mujtaba M.S., "Hybrid position / force control: a correct formulation", *The Int. J. of Robotics Research*, Vol. 11(4), 1992, p. 299-311.
- [**Fliess 95**] Fliess M., Lévine J., Martin P., Rouchon P., "Flatness and defect of nonlinear systems: introductory theory and examples", *Int. J. Control*, Vol. 61, 1995, p. 1327-1361.
- [**Forsythe 77**] Forsythe G.E., Malcom M.A., Moler C.B., *Computer methods for mathematical computations*, Prentice-Hall, Englewood Cliffs, 1977.
- [**Fournier 80**] Fournier A., "Génération de mouvements en robotique ; application des inverses généralisées et des pseudo-inverses", Thèse d'Etat, USTL, Montpellier, France, April 1980.
- [**Fourquet 90**] Fourquet J.-Y., "Mouvement en temps minimal pour les robots manipulateurs en tenant compte de leur dynamique", Thèse de Doctorat, UPS, Toulouse, France, 1990.
- [**Freidovich 97**] Freidovich L.B., Pervozvanski A.A., "Some estimates of performance for PID-like control of robotic manipulators", *Proc. IFAC Symp. on Robot Control, SYROCO'97*, Nantes, September 1997, p. 85-90.
- [**Freund 82**] Freund E., "Fast nonlinear control with arbitrary pole placement for industrial robots and manipulators", *The Int. J. of Robotics Research*, Vol. 1(1), 1982, p. 65-78.

[**Froissart 91**] Froissart C., "Génération adaptative de mouvement pour processus continus ; application au suivi de joint", Thèse de Doctorat, Université Pierre et Marie Curie, Paris, France, December 1991.

[FU 2007] FU S., Yao Y., Wu Y., "Comments on "A Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator" by B. Dasgupta and T.S. Mruthyunjaya [Mech. Mach. Theory 33 (1998) 1135-1152] ," Letter to the Editor / Mechanism and Machine Theory 42 (2007) 1668-1671.

[**Gaudin 92**] Gaudin H., "Contribution à l'identification *in situ* des constantes d'inertie et des lois de frottement articulaire d'un robot manipulateur en vue d'une application expérimentale au suivi de trajectoires optimales", Thèse de Doctorat, Université de Poitiers, France, November 1992.

[**Gautier 86**] Gautier M., "Identification of robots dynamics", *Proc. IFAC Symp. on Theory of Robots*, Vienna, Austria, December 1986, p. 351-356.

[**Gautier 88**] Gautier M., Khalil W., "On the identification of the inertial parameters of robots", *Proc. 27th IEEE Conf. on Decision and Control*, Austin, December 1988, p. 2264-2269.

[**Gautier 90a**] Gautier M., "Contribution à la modélisation et à l'identification des robots", Thèse de Doctorat d'Etat, ENSM, Nantes, France, May 1990.

[**Gautier 90b**] Gautier M., Khalil W., "Direct calculation of minimum set of inertial parameters of serial robots", *IEEE Trans. on Robotics and Automation*, Vol. RA-6(3), 1990, p. 368-373.

[**Gautier 91**] Gautier M., "Numerical calculation of the base inertial parameters", *J. of Robotic Systems*, Vol. 8(4), August 1991, p. 485-506.

[**Gautier 92a**] Gautier M., Khalil W., "Exciting trajectories for inertial parameters identification", *The Int. J. of Robotics Research*, Vol. 11(4), 1992, p. 362-375.

[**Gautier 92b**] Gautier M., "Optimal motion planning for robot's inertial parameters identification", *Proc. 31st IEEE Conf. on Decision and Control*, Tucson, December 1992, Vol. 1, p. 70-73.

[**Gautier 93**] Gautier M., Janin C., Pressé C., "On the validation of robot dynamic model", *Proc. 2nd European Control Conf., ECC'93*, Groningen, Holland, June-July 1993, p. 2291-2295.

[**Gautier 94**] Gautier M., Vandanjon P.-O., Pressé C., "Identification of inertial and drive gain parameters of robots", *Proc. IEEE 33th Conf. on Decision and Control*, Orlando, December 1994, p.3764-3769.

[**Gautier 95**] Gautier M., Khalil W., Restrepo P. P., "Identification of the dynamic parameters of a closed loop robot", *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, May 1995, p. 3045-3050.

[**Gautier 96**] Gautier M., "A comparison of filtered models for dynamic identification of robots", *Proc. IEEE 35th Conf. on Decision and Control*, Kobe, Japon, December 1996, p. 875-880.

[**Gautier 97**] Gautier M., "Dynamic identification of robots with power model", *Proc. Int. Conf. on Robotics and Automation*, Albuquerque, USA, avril 1997, p.1922-1927.

- [Gautier 2001] Gautier M., Khalil W., "Identification des paramètres des robots", Chapitre 4 of the book *Analyse et modélisation des robots manipulateurs*, ed. Dombre E., série IC2, Hermès Paris, 2001.
- [Gautier 2012] M. Gautier and S. Briot, "Global Identification of Drive Gains Parameters of Robots Using a Known Payload", *Proc. IEEE ICRA*, pp. 2812-2817, Saint Paul, MI, USA, 2012.
- [Geng 92] Geng, Z., Haynes, S., Lee, J. D. and Carrol, R. L., "On the dynamic model and kinematic analysis of a class of Stewart platforms", *Robotics and Autonomous Systems* **9**, p. 237-254, 1992.
- [Geffard 00] Geffard F., "Etude et conception de la commande en effort d'un télémanipulateur équipé d'un capteur d'effort à sa base et son extrémité", Thèse de Doctorat, Université de Nantes, France, December 2000.
- [Giordano 86] Giordano M., "Dynamic model of robots with a complex kinematic chain", *Proc. 16th Int. Symp. on Industrial Robots*, Bruxelles, September-October 1986, p. 377-388.
- [Gogu 2008] Gogu, G. *Structural Synthesis of Parallel Mechanisms, Part 1: Methodology*. Dordrecht: Springer Verlag, 2008.
- [Goldenberg 85] Goldenberg A.A., Benhabib B., Fenton R.G., "A complete generalized solution to inverse kinematics of robots", *IEEE J. of Robotics and Automation*, Vol. RA-1(1), 1985, p. 14-20.
- [Golub 83] Golub G.H., Van Loan C.F., *Matrix computations*, Johns Hopkins University Press, Baltimore, 1983.
- [Gorla 84] Gorla B., Renaud M., *Modèles des robots-manipulateurs ; application à leur commande*, Cepadues Editions, Toulouse, 1984.
- [Gosselin 88] Gosselin C., "Kinematic analysis, optimization and programming of parallel robotic manipulators", Ph. D. Thesis, McGill University, Montréal, Canada, June 1988.
- [Gosselin 90] Gosselin C., Angeles J., "Singularity analysis of closed-loop kinematic chains", *IEEE Trans. on Robotics and Automation*, Vol. RA-6(3), 1990, p. 281-290.
- [Gosselin 93] Gosselin C. M., "Parallel computational algorithms for the kinematics and dynamics of parallel manipulators", *IEEE Int. Conf. on Robotics and Automation*, Vol. 1, p.883-889, New York 1993.
- [Goswami 93] Goswami A., Quaid A., Peshkin M., "Identifying robot parameters using partial pose information", *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, Chicago, October 1993, p. 6-14.
- [Goudali 96] Goudali A., Lallemand J.-P., Zeghloul S., "Modeling of the 2-delta decoupled parallel robot", *Proc. 6th Int. Symp. on Robotics and Manufacturing, WAC'96*, Vol. 6, Montpellier, May 1996, p. 243-248.
- [Gough 56] Gough V.E., "Contribution to discussion of papers on research in automobile stability, control and tyre performance", *Proc. Auto. Div. Inst. Mech. Eng.*, 1956-1957.
- [Greville 60] Greville T.N., "Some applications of the pseudo-inverse of a matrix", *SIAM Review*, Vol. 2(1), 1960, p. 15-22.
- [Grudić 93] Grudić G.Z., Lawrence P.D., "Iterative inverse kinematics with manipulator configuration control", *IEEE Trans. on Robotics and Automation*, Vol. RA-9(4), August 1993, p. 476-483.

- [**Guglielmi 87**] Guglielmi M., Jonker E., Piasco J.-M., "Modelisation and identification of a two degrees of freedom SCARA robot using extended Kalman filtering", *Proc. Int. Conf. on Advanced Robotics, ICAR'87*, Versailles, October 1987, p. 137-148.
- [**Guyot 95**] Guyot G., "Contribution à l'étalonnage géométrique des robots manipulateurs", Thèse de Doctorat, Université de Nice-Sophia Antipolis, France, January 1995.
- [**Ha 89**] Ha I.J., Ko M.S., Kwon S.K., "An efficient estimation algorithm for the model parameters of robotic manipulators", *IEEE Trans. on Robotics and Automation*, Vol. RA-5(6), 1989, p. 386-394.
- [**Hahn 67**] Hahn W., *Stability of Motion*, Springer-Verlag, New York, 1967.
- [**Han 95**] Han K., Chung W.K., Youm Y., "Local structurization for the forward kinematics of parallel manipulators using extra sensor data", *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, May 1995 p. 514-520.
- [**Hayati 83**] Hayati S.A., "Robot arm geometric link parameter estimation", *Proc. 22nd IEEE Conf. Decision and Control*, San Antonio, December 1983, p. 1477-1483.
- [**Held 88**] Held V., Maron C., "Estimation of friction characteristics, inertial and coupling coefficients in robotic joints based on current and speed measurements", *Proc. IFAC Symp. on Robot Control, SYROCO'88*, 1988, p. 86.1-86.6.
- [**Hervé 78**] Hervé J.-M., "Analyse structurelle des mécanismes par groupe de déplacement", *J. of Mechanism and Machine Theory*, Vol. 13(4), 1978, p. 437-450.
- [**Hervé 91**] Hervé J.-M., Sparacino F., "Structural synthesis of parallel robot generating spatial translation", *Proc. Int. Conf. on Advanced Robotics, ICAR'91*, Pise, 1991, p. 808-813.
- [**Hoffman 79**] Hoffman R. and Hoffman, M., "Vibrational modes of an aircraft simulator motion system". In *Proc. 5th World Congress on Theory of Machines and Mechanisms*, p. 603-606, Montreal, July 1979.
- [**Hogan 85**] Hogan N., "Impedance control: an approach to manipulation", *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 107, March 1985, p. 1-24.
- [**Hogan 87**] Hogan N., "Stable execution of contact tasks using impedance control", *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, March-April 1987, p. 1047-1054.
- [**Hollerbach 80**] Hollerbach J.M., "An iterative lagrangian formulation of manipulators dynamics and a comparative study of dynamics formulation complexity", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-10(11), 1980, p. 730-736.
- [**Hollerbach 84a**] Hollerbach J.M., "Dynamic scaling of manipulator trajectories", *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 106(1), March 1984, p. 102-106.
- [**Hollerbach 84b**] Hollerbach J.M., "Optimum kinematic design for a seven degree of freedom manipulator", *Proc. 2nd Int. Symp. of Robotics Research*, Kyoto, August 1984, p. 349-356.
- [**Hollerbach 85**] Hollerbach J.M., Suh K.C., "Redundancy resolution of manipulators through torque optimization", *Proc. IEEE Int. Conf. on Robotics and Automation*, St Louis, March 1985, p. 1016-1021.
- [**Hollerbach 89**] Hollerbach J.M., "A survey of kinematic calibration", in *The Robotics Review n°1*, MIT Press Cambridge, 1989, p. 207-242.

- [**Hollerbach 95**] Hollerbach J.M., Lokhorst D., "Closed-loop kinematic calibration of the RSI 6-dof hand controller", *IEEE Trans. on Robotics and Automation*, Vol. RA-11, 1995, p. 352-359.
- [**Hollerbach 96**] Hollerbach J.M., Wampler C.W., "The calibration index and taxonomy of kinematic calibration methods", *The Int. J. of Robotics Research*, Vol. 14, 1996, p. 573-591.
- [**Hollerbach 2008**] Hollerbach J., Khalil W., Gautier M., "Model Identification", Chapter 14, Springer hand book of Robotics, Springer Verlag, ed Siciliano B, and Khatib O, 2008, pp. 321-344,
- [**Hooker 65**] Hooker W.W., Margulies G., "The dynamical attitude equations for a n-body satellite", *The Journal of the Astronautical Sciences*, Vol. 12(4), 1965, p. 123-128.
- [**Horowitz 80**] Horowitz R., Tomizuka M., "An adaptive control scheme for mechanical manipulators; compensation of non linearity and decoupling control", *Presentation at the Winter Meeting of ASME, Dynamic Systems and Control Division*, Chicago, 1980.
- [**Hsia 86**] Hsia T.C., "Adaptive control of robot manipulators; a review", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 183-189.
- [**Hsu 88**] Hsu P., Hauser J., Sastry S., "Dynamic control of redundant manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, Philadelphia, April 1988, p. 183-187.
- [**Hunt 83**] Hunt K. H., "Structural kinematics of in-parallel-actuated robot-arms", *Trans of ASME, J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 105, March 1983, p. 705-712.
- [**Husty 96**] Husty M., "An algorithm for solving the direct kinematics of Stewart-Gough-type platforms", *J. of Mechanisms and Machine Theory*, Vol. 31(4), 1996, p. 365-379.
- [**Ikits 97**] Ikits M., Hollerbach J.M., "Kinematic calibration using a plane constraint", *Proc. IEEE Int. Conf. on Robotics and Automation*, Albuquerque, April 1997, p. 3191-3196.
- [**Innocenti 93**] Innocenti C., Parenti-Castelli V., "Direct kinematics in analytical form of a general geometry 5-4 fully parallel manipulator", in *Computational kinematics*, J. Angeles et al. Eds., Klumer Academic Publishers, 1993, p. 141-152.
- [**Innocenti 95**] Innocenti C., "Analytical-form direct kinematics for the second scheme of a 5-5 general-geometry fully parallel manipulator", *J. of Robotic Systems*, Vol. 12(10), 1995, p. 661-676.
- [**Inoue 85**] Inoue H., Tsusaka Y., Fukuizumi T., "Parallel manipulator", *Proc. 3rd Int. Symp. of Robotics Research*, Gouvieux, October 1985, p. 69-75.
- [**Izaguirre 86**] Izaguirre A., Paul R.C.P., "Automatic generation of the dynamic equations of the robot manipulators using a LISP program", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 220-226.
- [**Ji 93**] Ji Z., "Study of the effect of leg inertia in Stewart platform", in *IEEE Int. Conf. on Robotics and Automation*, p. 121-126, Atlanta, May 1993.
- [**Judd 90**] Judd R., Knasinski A., "A technique to calibrate industrial robots with experimental verification", *IEEE Trans. on Robotics and Automation*, Vol. RA-6(1), 1990, p. 20-30.
- [**Kahn 69**] Kahn M.E., "The near minimum time control of open loop articulated kinematic chains", Ph. D. Thesis, Stanford University, Stanford, December 1969.

- [**Kanade 84**] Kanade T., Khosla P., Tanaka N., "Real-time control of the CMU direct-drive arm II using customized inverse dynamics", *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, December 1984, p. 1345-1352.
- [**Kane 83**] Kane T.R., Levinson D., "The use of Kane's dynamical equations in robotics", *The Int. J. of Robotics Research*, Vol. 2(3), 1983, p. 3-21.
- [**Kawasaki 88**] Kawasaki H., Nishimura K., "Terminal-link parameter estimation and trajectory control of robotic manipulators", *IEEE J. of Robotics and Automation*, Vol. RA-4(5), p. 485-490, 1988.
- [**Kawasaki 96**] Kawasaki H., Takahiro B., Kazuo K., "An efficient algorithm for the model-based adaptive control of robotic manipulators", *IEEE Trans. on Robotics and Automation*, Vol. RA-12(3), 1996, p. 496-501.
- [**Kazerooni 86**] Kazerooni H., Sheridan T., Houpt P., "Robust compliant motion for manipulators", Parts I and II, *IEEE J. of Robotics and Automation*, Vol. RA-2(2), 1986, p.83-105.
- [**Kazerounian 86**] Kazerounian K., Gupta K.C., "Manipulator dynamics using the extended zero reference position description", *IEEE J. of Robotics and Automation*, Vol. RA-2(4), 1986, p. 221-224.
- [**Kelly 88**] Kelly R., Ortega R., "Adaptive control of robot manipulators: an input-output approach", *Proc. IEEE Int. Conf. on Robotics and Automation*, Philadelphia, April 1988, p. 699-703.
- [**Kelly 95**] Kelly R., "A tuning procedure for stable PID control of robot manipulators", *Robotica*, Vol. 13, 1995, p. 141-148.
- [**Khalil 76**] Khalil W., "Modélisation et commande par ordinateur du manipulateur MA-23 ; extension à la conception par ordinateur des manipulateurs", Thèse de Docteur-Ingénieur, USTL, Montpellier, France, September 1976.
- [**Khalil 78**] Khalil W., "Contribution à la commande automatique des manipulateurs avec l'aide d'un modèle mathématique des mécanismes", Thèse d'Etat, USTL, Montpellier, France, October 1978.
- [**Khalil 79**] Khalil W., Liegeois A., Fournier A., "Commande dynamique des robots", *Revue RAIRO Automatique / Systems Analysis and Control*, Vol. 13(2), 1979, p. 189-201.
- [**Khalil 85a**] Khalil W., Kleinfinger J.-F., "Une modélisation performante pour la commande dynamique de robots", *Revue RAIRO, APII*, Vol. 6, 1985, p. 561-574.
- [**Khalil 85b**] Khalil W., Gautier M., "Identification of geometric parameters of robots", *Proc. IFAC Symp. on Robot Control, SYROCO'85*, Barcelone, November 1985, p. 191-194.
- [**Khalil 86a**] Khalil W., Kleinfinger J.-F., "A new geometric notation for open and closed-loop robots", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 1174-1180.
- [**Khalil 86b**] Khalil W., "On the explicit derivation of the inverse geometric models of robots", *Proc. IMACS-IFAC Symp.*, Villeneuve d'Ascq, June 1986, p. 541-546.
- [**Khalil 86c**] Khalil W., Kleinfinger J.-F., Gautier M., "Reducing the computational burden of the dynamic model of robots", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 525-531.

- [**Khalil 87a**] Khalil W., Chevallereau C., "An efficient algorithm for the dynamic control of robots in the cartesian space", *Proc. 26th IEEE Conf. on Decision and Control*, Los Angeles, December 1987, p. 582-588.
- [**Khalil 87b**] Khalil W., Kleinfinger J.-F., "Minimum operations and minimum parameters of the dynamic model of tree structure robots", *IEEE J. of Robotics and Automation*, Vol. RA-3(6), December 1987, p. 517-526.
- [**Khalil 89a**] Khalil W., Bennis F., Chevallereau C., Kleinfinger J.-F., "SYMORO: a software package for the symbolic modelling of robots", *Proc. 20th Int. Symp. on Industrial Robots*, Tokyo, October 1989, p. 1023-1030.
- [**Khalil 89b**] Khalil W., Bennis F., Gautier M., "Calculation of the minimum inertial parameters of tree structure robots", *Proc. Int. Conf. on Advanced Robotics, ICAR'89*, Columbus, USA, Springer-Verlag, New York, 1989, p. 189-201.
- [**Khalil 89c**] Khalil W., Caenen J.-L., Enguehard C., "Identification and calibration of the geometric parameters of robots", *Proc. 1st Experimental Robot Conference*, Montreal, 1989, Springer-Verlag, New York, Vol. 139, p. 528-538.
- [**Khalil 90a**] Khalil W., Bennis F., Gautier M., "The use of the generalized links to determine the minimum inertial parameters of robots", *J. of Robotic Systems*, Vol. 7(2), 1990, p. 225-242.
- [**Khalil 90b**] Khalil W., Bennis F., "Calcul de la matrice d'inertie des robots à chaîne ouverte simple ou arborescente", Rapport interne n° 90-09, LAN-ENSM, April 1990.
- [**Khalil 90c**] Khalil W., Bennis F., "Automatic generation of the inverse geometric model of robots", *J. of Robotics and Autonomous Systems*, Vol. 7, 1991, p. 1-10.
- [**Khalil 91a**] Khalil W., Gautier M., "Calculation of the identifiable parameters for robot calibration", *Proc. IFAC Symp. on Identification and System Parameter Estimation*, Budapest, 1991, p. 888-892.
- [**Khalil 91b**] Khalil W., Gautier M., Enguehard C., "Identifiable parameters and optimum configurations for robot calibration", *Robotica*, Vol. 9, 1991, p. 63-70.
- [**Khalil 93**] Khalil W., Gautier M., "Computed current control of robots", *Proc. IFAC 12th World Congress*, Sydney, Australia, July 1993, Vol. IV, p. 129-132.
- [**Khalil 94a**] Khalil W., Bennis F., "Comments on Direct Calculation of Minimum Set of Inertial Parameters of Serial Robots", *IEEE Trans. on Robotics and Automation*, Vol. RA-10(1), 1994, p. 78-79.
- [**Khalil 94b**] Khalil W., Murareci D., "On the general solution of the inverse kinematics of six-degrees-of-freedom manipulators", *Proc. Int. Workshop on Advances in Robot Kinematics, ARK 94*, Slovénie, July 1994.
- [**Khalil 95a**] Khalil W., Bennis F., "Symbolic calculation of the base inertial parameters of closed-loop robots", *The Int. J. of Robotics Research*, Vol. 14(2), April 1995, p. 112-128.
- [**Khalil 95b**] Khalil W., Garcia G., Delagarde J.-F. "Calibration of the geometric parameters of robots without external sensors", *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, May 1995, p. 3039-3044.
- [**Khalil 96a**] Khalil W., Restrepo P.P., "An efficient algorithm for the calculation of the filtered dynamic model of robots", *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, April 1996, p. 323-329.

- [**Khalil 96b**] Khalil W., Lemoine P., Gautier M., "Autonomous calibration of robots using planar points", *Proc. 6th Int. Symp. on Robotics and Manufacturing, WAC'96*, Vol. 3, Montpellier, May 1996, p. 383-388.
- [**Khalil 96c**] Khalil W., Murareci D., "Kinematic analysis and singular configuration of a class of parallel robots", *J. of Mathematic and Computer in Smulation*, n°1245, 1996, p. 1-14.
- [**Khalil 97a**] Khalil W., Creusot D., "SYMORO+: a system for the symbolic modelling of robots", *Robotica*, Vol. 15, 1997, p. 153-161.
- [**Khalil 97b**] Khalil W., Murareci D., "Autonomous calibration of parallel robots", *Proc. IFAC Symp. on Robot Control, SYROCO'97*, Nantes, September 1997, p. 425-430.
- [**Khalil 99**] Khalil W., Lemoine Ph., "Gecaro: A system for the geometric calibration of robots", *APII-Jesa*, Vol.33, n°5-6 July, 1999, p.717-739.
- [**Khalil 99a**] Khalil W., Lemoine P., "Gecaro: a system for the geometric calibration of robots", *Revue APII-JESA*, Vol. 33(5-6), 1999, p. 717-739.
- [**Khalil 99b**] Khalil W., Besnard S., "Self calibration of Stewart-Gough parallel robots without extra sensors", *IEEE Trans. on Robotics and Automation*, Vol. RA-15(6), p. 1116-1121, December 1999.
- [**Khalil 00a**] Khalil W., Gautier M., "Modeling of mechanical systems with lumped elasticity", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 2000, p. 3965-3970.
- [**Khalil 00b**] Khalil W., Besnard S., Lemoine P., "Comparison study of the geometric parameters calibration methods ", *Int. J. of Robotics and Automation*, Vol. 15(2), 2000, p. 56-67.
- [**Khalil 03**] Khalil W., ed. "Commande des robots", Série IC2, Hermès, 2002.
- [**Khalil 2002**] Khalil W. Dombre E., *Modeling, identification and control of robots*, Hermes Penton, London, 2002.
- [**Khalil 2004**] Khalil W. and Guegan S., "Inverse and Direct Dynamic Modeling of Gough-Stewart Robots", *IEEE Transactions on Robotics and Automation*, 20(4), p. 754-762, 2004.
- [**Khalil 2007**] Khalil W., Ibrahim O., "General solution for the Dynamic modeling of parallel robots", *Journal of Intelligent and Robotic Systems*, Vol.49, 2007, p.19-37.
- [**Khalil 2007a**] W.Khalil W., Gautier M., Lemoine Ph., " Identification of the payload inertial parameters of industrial manipulators", ICRA 07, Rome, April 2007
- [**Khalil 2007**] Khalil W., G. Gallot G., Boyer F., "Dynamic Modeling and Simulation of a 3-D Serial Eel-Like Robot", *IEEE Transactions on Systems, Man and Cybernetics, Part C: Application and reviews*, Vol. 37, N° 6, November 2007.
- [**Khalil 2010**] W.Khalil, « Dynamic modeling of Robots using Newton-Euler Formulation», Lecture Notes in Electrical Engineering 89, Informatics in Control, Automation and Robotics, J.A.Cetto, J-L Ferrier, J. Filipe (editors), 2010, p. 3-20, Springer.
- [**Khatib 80**] Khatib O., "Commande dynamique dans l'espace opérationnel des robots-manipulateurs en présence d'obstacles", Thèse de Docteur-Ingénieur, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France, December 1980.

- [**Khatib 87**] Khatib O., "A unified approach for motion and force control of robot manipulators: the operational space formulation", *IEEE J. of Robotics and Automation*, Vol. RA-3(1), February 1987, p. 43-53.
- [**Khelfi 95**] Khelfi M.-F., "Observateurs non linéaires : application à la commande des robots manipulateurs", Thèse de Doctorat, Université Poincaré Nancy I, France, 1995.
- [**Kholi 85**] Kholi D., Spanos J., "Workspace analysis of mechanical manipulators using polynomial discriminants", *J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 107, June 1985, p. 209-215.
- [**Kholi 87**] Kholi D., Hsu M.S., "The jacobian analysis of workspaces of mechanical manipulators", *J. of Mechanisms and Machine Theory*, Vol. 22(3), 1987, p. 265-275.
- [**Khosla 85**] Khosla P.K., Kanade T., "Parameter identification of robot dynamics", *Proc. 24th IEEE Conf. on Decision and Control*, Fort-Lauderdale, December 1985, p. 1754-1760.
- [**Khosla 86**] Khosla P.K., "Real-time control and identification of direct drive manipulators", Ph. D. Thesis, Carnegie Mellon University, Pittsburgh, 1986.
- [**Kircanski 85**] Kircanski M., Vukobratovic M., "Computer-aided generation of manipulator kinematic models in symbolic form", *Proc. 15th Int. Symp. on Industrial Robots*, Tokyo, September 1985, p. 1043-1049.
- [**Klein 83**] Klein C.A., Huang C., "Review of pseudo inverse control for use with kinematically redundant manipulators", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-13(2), 1983, p. 245-250.
- [**Klein 84**] Klein C.A., "Use of redundancy in the design of robotic systems", *Proc. 2nd Int. Symp. of Robotic Research*, Kyoto, August 1984, p. 58-65.
- [**Klein 95**] Klein C.A., Chu-Jenq, Ahmed S., "A new formulation of the extended Jacobian method and its use in mapping algorithmic singularities for kinematically redundant manipulators", *IEEE Trans. on Robotics and Automation*, Vol. RA-11(1), 1995, p. 50-55.
- [**Kleinfinger 86a**] Kleinfinger J.-F., "Modélisation dynamique de robots à chaîne cinématique simple, arborescente ou fermée, en vue de leur commande", Thèse de Doctorat, ENSM, Nantes, France, May 1986.
- [**Kleinfinger 86b**] Kleinfinger J.-F., Khalil W., "Dynamic modelling of closed-chain robots", *Proc. 16th Int. Symp. on Industrial Robots*, Bruxelles, September-October 1986, p. 401-412.
- [**Klema 80**] Klema V.C., Lanio A.J., "The singular value decomposition: its computation and some applications", *IEEE Trans. on Automatic Control*, Vol. AC-25(2), 1980, p. 164-176.
- [**Koditschek 84**] Koditschek D.E., "Natural motion for robot arms", *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, December 1984, p. 737-735.
- [**Konstantinov 81**] Konstantinov M.S., Markov M.D., Nenchev D.N., "Kinematic control of redundant manipulators", *Proc. 11th Int. Symp. on Industrial Robots*, Tokyo, October 1981, p. 561-568.
- [**Korrami 88**] Korrami F., Özgünür U., "Decentralized control of robot manipulators via state and proportional-integral feedback", *Proc. IEEE Int. Conf. on Robotics and Automation*, Philadelphia, April 1988, p. 1198-1203.
- [**Kreuzer 79**] Kreuzer E.J., "Dynamical analysis of mechanisms using symbolical equation manipulation", *Proc. 5th World Congress on Theory of Machines and Mechanisms*, Montréal, 1979, p. 599-602.

- [Landau 79] Landau I. D., *Adaptive control; The model reference approach*, Marcel Dekker Inc. New York, 1979.
- [Landau 88] Landau I. D., Horowitz R., "Synthesis of adaptive controllers for robot manipulators using a passive feedback system approach", *Proc. IEEE Int. Conf. on Robotics and Automation*, Philadelphia, April 1988, p. 1028-1033.
- [Lavallée 92] Lavallée S., "Procédé d'étalonnage d'un robot", Brevet n° FR 2 696 969 du 21/10/92.
- [Lawson 74] Lawson C.L., Hanson R.J., *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, 1974.
- [Lazard 92] Lazard D., "Stewart platforms and Grubner basis", *Proc. 3rd Int. Workshop on Advances in Robot Kinematics*, ARK 92, 1992, p. 136-142.
- [Le Borzec 75] Le Borzec R., Lotterie J., *Principes de la théorie des mécanismes*, Dunod, 1975.
- [Lebret 93] Lebret G., Liu G. K., Lewis, F. L., "Dynamic analysis and control of a Stewart platform manipulator", *J. of Robotic Systems* **10**(5), p. 629-655, July 1993.
- [Lee 83] Lee C.S.G., Ziegler M., "A geometric approach in solving the inverse kinematics of PUMA robots", *Proc. 13th Int. Symp. on Industrial Robots*, Chicago, April 1983, p. (16-1)-(16-18).
- [Lee 88] Lee H.Y., Liang C.G., "Displacement analysis of the general 7-link 7R mechanism", *J. of Mechanism and Machine Theory*, Vol. 23(3), 1988, p. 219-226.
- [Lee 88b] Lee K. M., Shah D. K., "Dynamic analysis of a three-degrees-of-freedom in-parallel actuated manipulator", *IEEE J. of Robotics and Automation* 4(3), p.361-368, June 1988.
- [Lee 93] Lee H.Y., Roth B., "A closed-form solution of the forward displacement analysis of a class of in-parallel robots", *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, May 1993, p. 720-724.
- [Léon 91] Léon J.-C., *Modélisation et construction de surfaces pour la CFAO*, Hermès, Paris, 1991.
- [Lewis 93] Lewis F.L., Abdallah C.T., Dawson D.M., *Control of robot manipulators*, Macmillan, New York, 1993.
- [Li 89] Li W., Slotine J.-J.E., "An indirect adaptive robot controller", *Systems & Control Letters*, Vol. 12, 1989, p. 259-266.
- [Liégeois 79] Liégeois A., Dombre E., "Analyse des robots industriels ; relations entre structures, performances et fonctions", Rapport IRIA n° 79102, Projet SURF, LAM, Montpellier, 1979.
- [Liu 2000] Liu M.-J., Li C.-X., Li C.-N., "Dynamics analysis of the Gough-Stewart platform manipulator", *IEEE Transaction on Robotics and Automation*, **16**(1), p. 94-98 (February 2000).
- [Lilly 90] Lilly K.W., Orin D.E., "Efficient O(N) computation of the operational space inertia matrix", *Proc. IEEE Int. Conf. on Robotics and Automation*, Cincinnati, May 1990, p. 1014-1019.

- [**Lin 83**] Lin C.S., Chang P.R., Luh J.Y.S., "Formulation and optimization of cubic polynomial joint trajectories for industrial robots", *IEEE Trans. on Automatic Control*, Vol. AC-28(12), December 1983, p. 1066-1073.
- [**Lin 92**] Lin W., Griffiths M., Duffy J., "Forward displacement analysis of the 4-4 Stewart platforms", *Trans. of the ASME, J. of Mechanical Design*, Vol. 114, September 1992, p. 444-450.
- [**Llibre 83**] Llibre M., Mampey R., Chrétien J.P., "Simulation de la dynamique des robots manipulateurs motorisés", *Congrès AFCET : Productique et Robotique Intelligente*, Besançon, November 1983, p. 197-207.
- [**Lloyd 96**] Lloyd J.E., "Using Puiseux series to control non-redundant robots at singularities", *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, April 1996, p. 1877-1882.
- [**Lu 93**] Lu Z., Shimoga K.B., Goldberg A., "Experimental determination of dynamic parameters of robotic arms", *J. of Robotic Systems*, Vol. 10(8), 1993, p. 1009-1029.
- [**Luh 80a**] Luh J.Y.S., Walker M.W., Paul R.C.P., "Resolved-acceleration control of mechanical manipulators", *IEEE Trans. on Automatic Control*, Vol. AC-25(3), June 1980, p. 468-474.
- [**Luh 80b**] Luh J.Y.S., Walker M.W., Paul R.C.P., "On-line computational scheme for mechanical manipulators", *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 102(2), 1980, p. 69-76.
- [**Luh 85a**] Luh J.Y.S., Gu Y.L., "Industrial robots with seven joints", *Proc. IEEE Int. Conf. on Robotics and Automation*, St Louis, March 1985, p. 1010-1015.
- [**Luh 85b**] Luh J.Y.S., Zheng Y.F., "Computation of input generalized forces for robots with closed kinematic chain mechanisms", *IEEE J. of Robotics and Automation*, Vol. RA-1(2), 1985, p. 95-103.
- [**Ma 91**] Ma O., Angeles J., "Architecture singularities of platform manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, Sacramento, April 1991, p. 1542-1547.
- [**Maciejewski 85**] Maciejewski A.A., Klein C.A., "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments", *The Int. J. of Robotics Research*, Vol. 4(3), Fall 1985, p. 109-117.
- [**Maciejewski 88**] Maciejewski A.A., Klein C.A., "Numerical filtering operation of robotic manipulators through kinematically singular configurations", *J. of Robotic Systems*, Vol. 5(6), 1988, p. 527-552.
- [**Maciejewski 89**] Maciejewski A.A., Klein C.A., "The singular value decomposition: computation and applications to robotics", *The Int. J. of Robotics Research*, Vol. 8(6), 1989, p. 63-79.
- [**Manaoui 85**] Manaoui O., "Calcul automatique de transformateurs de coordonnées analytiques de robots à partir de classes de solutions préétablies", Rapport de DEA, USTL, Montpellier, July 1985.
- [**Mansard 2006**] Mansard N. Enchaînement de tâches robotiques. PhD thesis, Université de Rennes 1, Mention informatique, December 2006.
- [**Mansard 2007**] Mansard N., Chaumette, "Task sequencing for sensor-based control," *IEEE Trans. Robotics*, vol. 23, no. 1, pp. 60-72, Feb. 2007.

- [**Mason 82**] Mason M.T., "Compliant motion", in *Robot motion: planning and control*, M. Brady *et al.* Eds., MIT Press, Cambridge, 1982.
- [**Maurine 96**] Maurine P., "Développement et mise en œuvre de méthodologies d'étalonnage de robots manipulateurs industriels", Thèse de Doctorat, Université Montpellier II, France, December 1996.
- [**Mavroidis 93**] Mavroidis C., "Résolution du problème géométrique inverse pour les manipulateurs série à six degrés de liberté", Thèse de Doctorat, Université Pierre et Marie Curie, Paris, France, May 1993.
- [**Mayeda 84**] Mayeda H., Osuka K., Kangawa A., "A new identification method for serial manipulator arms", *Proc. IFAC 9th World Congress*, Budapest, July 1984, p. 74-79.
- [**Mayeda 90**] Mayeda H., Yoshida K., Osuka K., "Base parameters of manipulator dynamic models", *IEEE Trans. on Robotics and Automation*, Vol. RA-6(3), 1990, p. 312-321.
- [**Megahed 82**] Megahed S., Renaud M., "Minimization of the computation time necessary for the dynamic control", *Proc. 12th Int. Symp. on Industrial Robots*, Paris, June 1982, p. 469-478.
- [**Megahed 84**] Megahed S., "Contribution à la modélisation géométrique et dynamique des robots manipulateurs ayant une structure de chaîne cinématique simple ou complexe ; application à leur commande", Thèse d'Etat, UPS, Toulouse, France, July 1984.
- [**Mendel 73**] Mendel J.M., *Discrete techniques of parameter estimation: the equation error formulation*, Marcel Dekker Inc. New York, 1973.
- [**Merlet 86**] Merlet J.-P., "Contribution à la formalisation de la commande par retour d'effort en robotique ; application à la commande de robots parallèles", Thèse de Doctorat, Université Pierre et Marie Curie, Paris, France, June 1986.
- [**Merlet 89**] Merlet J.-P., "Singular configurations of parallel manipulators and Grassmann geometry", *The Int. J. of Robotics Research*, Vol. 8(5), October 1989, p. 45-56.
- [**Merlet 93**] Merlet J.-P., "Closed-form resolution of the direct kinematics of parallel manipulators using extra sensor data", *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, May 1993, p. 200-204.
- [**Merlet 2000**] Merlet J.-P., *Parallel robots*, Kluwer Academic Publ., Dordrecht, The Netherlands, 2000.
- [**Middleton 88**] Middleton R.H., Goodwin G.C., "Adaptive computed torque control for rigid link manipulators", *Systems & Control Letters*, Vol. 10, 1988, p. 9-16.
- [**Milenkovic 83**] Milenkovic V., Huang B., "Kinematics of major robot linkage", *Proc. 13th Int. Symp. on Industrial Robots*, Chicago, April 1983, p. 16.31-16.47.
- [**Miller 2004**] Miller K., "Optimal Design and Modeling of Spatial Parallel Manipulators", *The Int. J. of Robotics Research*, Vol. 23(2), p.127-140 (February 2004).
- [**Mooring 91**] Mooring B.W., Roth Z.S., Driels M.R., *Fundamentals of manipulator calibration*, John Wiley & Sons, New York, 1991.
- [**Morel 94**] Morel G., "Programmation et adaptation d'impédance de manipulateurs au contact", Thèse de Doctorat, Université Pierre et Marie Curie, Paris, France, June 1994.

- [**Moroskine 1954**] Moroskine, Y. F. "General analysis of the theory of mechanisms." *Akad Nauk SSSR, Trudy Sem, Teorii Masin i Mekhanizmov*, 1954, vol. 14, p. 25-50.
- [**Moroskine 1958**] Moroskine, Y. F. "On the geometry of complex kinematic chains." *Sov Phys-Dokl.*, 1958, vol. 3, n° 2, p. 269-272.
- [**Murareci 97**] Murareci D., "Contribution à la modélisation géométrique et à l'étalonnage des robots séries et parallèles", Thèse de Doctorat, Université de Nantes et Ecole Centrale de Nantes, France, March 1997.
- [**Murphy 93**] Murphy S.H., Wen J.T.U., "Analysis of active manipulator elements in space manipulation", *IEEE Trans. on Robotics and Automation*, Vol. RA-9(5), October 1993, p. 544-552.
- [**Murray 84**] Murray J.J., Newman C.P., "ARM: an algebraic robot dynamic modeling program", *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, March 1984, p. 103-104.
- [**Nahvi 94**] Nahvi A., Hollerbach J.M., Hayward V., "Calibration of parallel robot using multiple kinematic closed loops", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Diego, May 1994, p. 407-412.
- [**Nakamura 86**] Nakamura Y., Hanafusa Y., "Inverse kinematic solutions with singularity robustness for robot manipulator control", *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 108, 1986, p. 163-171.
- [**Nakamura 87**] Nakamura Y., Hanafusa Y., Yoshikawa T., "Task-priority based redundancy control of a robot manipulator", *The Int. J. of Robotics Research*, Vol. 6(2), 1987, p. 3-15.
- [**Nanua 90**] Nanua P., Waldron K.J., Murthy V., "Direct kinematic solution of a Stewart platform", *IEEE Trans. on Robotics and Automation*, Vol. RA-6(4), 1990, p. 438-444.
- [**Nenchev 92**] Nenchev D.N., "Restricted jacobian matrices of redundant manipulators in constrained motion tasks", *The Int. J. of Robotics Research*, Vol. 11(6), 1992, p. 584-597.
- [**Nevins 73**] Nevins J.L., Whitney D.E., "The force vector assembler concept", *Proc. 1st CISM-IFTOMM Symp. on Theory and Practice of Robots and Manipulators*, Udine, September 1973, p. 273-288.
- [**Nevins 77**] Nevins J.L. *et al.*, "Exploratory research in industrial modular assembly", Charles Stark Draper Lab., Report R-1111, 1977.
- [**Newman 79**] Newman W.H., Sproull R.F., *Principles of interactive computer graphics*, McGraw Hill, New York, 1979.
- [**Nicosia 84**] Nicosia S., Tomei P., "Model reference adaptive control algorithms for industrial robots", *Automatica*, Vol. 20(5), 1984, p. 635-644.
- [**Nicosia 90**] Nicosia S., Tomei P., "Robot control by using only joint position measurements", *IEEE Trans. on Automatic Control*, Vol. AC-35(5), 1990, p. 1058-1061.
- [**Nielsen 91**] Nielsen L., Canudas de Wit C., Hagander P., "Controllability issues of robots near singular configurations", in *Advanced Robot Control, Lecture Notes in Control and Information Sciences*, Springer-Verlag, New York, 1991, p. 307-314.
- [**Olsen 86**] Olsen H.B., Bekey G.A., "Identification of robot dynamics", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, April 1986, p. 1004-1010.

- [**Orin 79**] Orin D.E., McGhee R.B., Vukobratovic M., Hartoch G., "Kinematic and kinetic analysis of open-chain linkages utilizing Newton-Euler methods", *Mathematical Biosciences*, Vol. 43, 1979, p. 107-130.
- [**Ortega 89**] Ortega R., Spong M.W., "Adaptive motion control of rigid robots: a tutorial", *Automatica*, Vol. 25(6), 1989, p. 877-888.
- [**Paden 88**] Paden B., Panja R., "Globally asymptotically stable 'PD+' controller for robot manipulator", *Int. J. Control*, Vol. 47, 1988, p. 877-888.
- [**Paul 72**] Paul R.C.P., "Modelling, trajectory calculation, and servoing of a computer controlled arm", Ph. D. Thesis, Stanford Artificial Intelligence Lab., Stanford, 1972.
- [**Paul 81**] Paul R.C.P., *Robot manipulators: mathematics, programming and control*, MIT Press, Cambridge, 1981.
- [**Payannet 85**] Payannet D., "Modélisation et correction des erreurs statiques des robots manipulateurs", Thèse de Doctorat, USTL, Montpellier, France, 1985.
- [**Penrose 55**] Penrose R., "A generalized inverse for matrices", *Proc. Cambridge Philos. Soc.*, Vol. 51, 1955, p. 406-413.
- [**Perdereau 91**] Perdereau V., "Contribution à la commande hybride force-position", Thèse de Doctorat, Université Pierre et Marie Curie, Paris, France, February 1991.
- [**Pham 91a**] Pham C., Chedmail P., Gautier M., "Determination of base parameters of flexible links manipulators", *Proc. IMACS MCTS Symposium, Modelling and Control of Technological Systems*, Lille, May 1991, Vol. 1, p. 524-529.
- [**Pham 91b**] Pham C., Gautier M., "Essential parameters of robots", *Proc. 30th IEEE Conf. on Decision and Control*, Brighton, December 1991, p. 2769-2774.
- [**Pham 2001**] Pham M.T., Gautier M., Poignet P., "Identification of joint stiffness with bandpass filtering", *Int. Conf. on Robotics and Automation*, Seoul, Korea, mai 2001.
- [**Pieper 68**] Pieper D.L., "The kinematics of manipulators under computer control", Ph. D. Thesis, Stanford University, 1968.
- [**Pierrot 91a**] Pierrot F., "Robots pleinement parallèles légers : conception, modélisation et commande", Thèse de Doctorat, Université Montpellier II, France, April 1991.
- [**Pierrot 91b**] Pierrot F., Fournier A., Dauchez P., "Towards a fully-parallel six dof robot for high-speed applications", *Proc. IEEE Int. Conf. on Robotics and Automation*, Sacramento, April 1991, p. 1288-1293.
- [**Pledel 96**] Pledel P., "Génération de mouvements optimaux pour un robot manipulateur", Thèse de doctorat, Université de Nantes et Ecole Centrale de Nantes, France, September 1996.
- [**Poignet 2000**] Poignet P., Gautier M., "Comparison of weighted least squares and extended kalman filtering methods for dynamic identification of robots", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, 2000, p. 3622-3627.
- [**Popov 73**] Popov V.M., *Hyperstability of control systems*, Springer-Verlag, New York, 1973.
- [**Potkonjak 86**] Potkonjak V., "Thermal criterion for the selection of DC drives for industrial robots", *Proc. 16th Int. Symp. on Industrial Robots*, Bruxelles, September-October 1986, p. 129-140.

- [**Powell 64**] Powell, M.J.D., "An efficient method for finding the minimum of a function of several variables without calculating derivatives", *The Computer Journal*, Vol. 7, 1964, p. 155-162.
- [**Pressé 93**] Pressé C., Gautier M., "New criteria of exciting trajectories for robot identification", *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, May 1993, p. 907-912.
- [**Pressé 94**] Pressé C., Identification des paramètres dynamiques des robots", Thèse de Doctorat, Université de Nantes et Ecole Centrale de Nantes, France, October 1994.
- [**Priel 90**] Priel M., *Les robots industriels : caractéristiques, performance et choix*, Collection AFNOR Technique, 1990.
- [**Prüfer 94**] Prüfer M., Schmidt C., Wahl F., "Identification of robots dynamics with differential and integral models: a comparison", *Proc. IEEE Int. Conf. on Robotics and Automation*, San Diego, May 1994, p. 340-345.
- [**Pujas 95**] Pujas A., "Etude de la robustesse de schémas de commande position / force pour robots à deux bras", Thèse de Doctorat, Université Montpellier II, France, June 1995.
- [**Qu 91**] Qu Z., Dorsey J., "Robust PID control of robots", *Int. J. Robotics and Automation*, Vol. 6(4), 1991, p. 228-235.
- [**Raghavan 90**] Raghavan M., Roth B., "Inverse kinematics of the general 6R manipulator and related linkages", *Trans. of the ASME, J. of Mechanical Design*, Vol. 115, 1990, p. 502-508.
- [**Raghavan 91**] Raghavan M., "The Stewart platform of general geometry has 40 configurations", in *Advances in Design Automation*, ASME Press, New-York, 1991, p. 397-402.
- [**Raibert 77**] Raibert M.H., "Analytic equations vs. table look-up for manipulation: a unifying concept", *Proc. 16th IEEE Conf. on Decision and Control*, New Orleans, 1977, p. 576-579.
- [**Raibert 78**] Raibert M.H., Horn B.K.P., "Manipulator control using the configuration space method", *The Industrial Robot*, Vol. 5(2), 1978, p. 69-73.
- [**Raibert 81**] Raibert M.H., Craig J.J., "Hybrid position/force control of manipulators", *Trans. of the ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 103, June 1981, p. 126-133.
- [**Raucent 90**] Raucent B., "Identification des paramètres dynamiques des robots manipulateurs", Thèse de Doctorat, Université de Louvain, Belgium, 1990.
- [**Raucent 92**] Raucent B., Bastin G., Campion G., Samin J.-C., Willems P.Y., "Identification of barycentric parameters of robotic manipulators from external measurements", *Automatica*, Vol. 28(5), 1992, p. 1011-1016.
- [**Reboulet 85**] Reboulet C., Robert A., "Hybrid control of a manipulator equipped with an active compliant wrist", *Proc. 3rd Int. Symp. of Robotics Research*, Gouvieux, October 1985, p. 76-80.
- [**Reboulet 88**] Reboulet C., "Modélisation des robots parallèles", in *Techniques de la robotique : architectures et commandes*, Hermès, Paris, 1988.
- [**Reboulet 91**] Reboulet C., Berthomieu T., "Dynamic models of a six degree of freedom parallel manipulators", in *proceeding ICAR*, p.1153-1157, Pise, June 1991.
- [**Renaud 75**] Renaud M., "Contribution à l'étude de la modélisation et de la commande des

systèmes mécaniques articulés", Thèse de Docteur-Ingénieur, UPS, Toulouse, France, December 1975.

[**Renaud 80a**] Renaud M., "Contribution à la modélisation et à la commande dynamique des robots manipulateurs", Thèse d'Etat, UPS, Toulouse, France, September 1980.

[**Renaud 80b**] Renaud M., "Calcul de la matrice jacobienne nécessaire à la commande coordonnée d'un manipulateur", *J. of Mechanism and Machine Theory*, Vol. 15(1), 1980, p. 81-91.

[**Renaud 85**] Renaud M., "A near minimum iterative analytical procedure for obtaining a robot-manipulator dynamic model", *IUTAM/IFToMM Symp. on Dynamics of Multi-body Systems*, Udine, 1985.

[**Renaud 87**] Renaud M., "Quasi-minimal computation of the dynamic model of a robot manipulator utilizing the Newton-Euler formalism and the notion of augmented body", *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, March-April 1987, p. 1677-1682.

[**Restrepo 95**] Restrepo P.P., Gautier M., "Calibration of drive chain of robot joints", *Proc. 4th IEEE Conf. on Control Applications, CCA'95*, Albany, 1995, p. 526-531.

[**Restrepo 96**] Restrepo P.P., Contribution à la modélisation, identification et commande des robots à structures fermées : application au robot Acma SR400", Thèse de Doctorat, Université de Nantes et Ecole Centrale de Nantes, France, October 1996.

[**Richalet 98**] Richalet J., *Pratique de l'identification*, 2^{ième} édition, Hermès, Paris, 1998.

[**Robert 86**] Robert A., "Commande hybride position-force ; mise en œuvre et expérimentation sur un micro-ordinateur parallèle", Thèse de Docteur-Ingénieur, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France, December 1986.

[**Roberts 65**] Roberts L.G., "Homogeneous matrix representation and manipulation of N-dimensional constructs", MIT Lincoln Lab., MS 1405, May 1965.

[**Rocco 96**] Rocco P., "Stability of PID control for industrial robot arms", *IEEE Trans. on Robotics and Automation*, Vol. RA-12(4), 1996, p. 607-614.

[**Roth 76**] Roth B., "Performance evaluation of manipulators from a kinematic viewpoint", Cours de Robotique, IRIA, Toulouse, 1976, p. 233-263.

[**Roth 87**] Roth Z.S., Mooring B.W., Ravani B., "An overview of robot calibration", *IEEE J. of Robotics and Automation*, Vol. RA-3(5), October 1987, p. 377-385.

[**Sadegh 87**] Sadegh N., Horowitz R., "Stability analysis of an adaptive controller for robotic manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, March-April 1987, p. 1223-1229.

[**Sadegh 90**] Sadegh N., Horowitz R., "Stability and robustness analysis of a class of adaptive controllers for robotic manipulators", *The Int. J. of Robotics Research*, Vol. 9(3), 1990, p. 74-92.

[**Salisbury 80**] Salisbury J.K., "Active stiffness control of a manipulator in cartesian coordinates", *Proc. 19th IEEE Conf. on Decision and Control*, Albuquerque, December 1980, p. 95-100.

[**Salmon 1885**] Salmon G., *Lessons introductory to the modern higher algebra*, Chelsea Publishing CO., New York, 1885.

[**Samson 83**] Samson C., "Problèmes en identification et commande de systèmes dynamiques", Thèse d'Etat, Rennes, France, 1983.

- [**Samson 87**] Samson C., "Robust control of a class of non-linear systems and applications to robotics", *Int. J. of Adaptive Control and Signal Processing*, Vol. 1, 1987, p. 49-68.
- [**Samson 91**] Samson C., Le Borgne M., Espiau B., *Robot Control*, Oxford University Press, Oxford, 1991.
- [**Schefer 82**] Schefer B., "Geometric control and calibration method of an industrial robot", *Proc. 12th Int. Symposium on Industrial Robotics*, Paris, 1982, p. 331-339.
- [**Sciavicco 94**] Sciavicco L., Siciliano B., Villani L., "On dynamic modelling of gear-driven rigid robot manipulators", *Proc. 4th IFAC Symp. on Robot Control, SYROCO'94*, Capri, September 1994, p. 543-549.
- [**Sgarbi 92**] Sgarbi F., Cammoun R., "Real time trajectory generation using filtering techniques", *Proc. 2nd Int. Conf. on Automation, Robotics, and Computer Vision, ICARCV'92*, Singapore, September 1992, p. RO-8.5.1-RO-8.5.6.
- [**Sheth 71**] Sheth P.N., Uicker J.J., "A generalized symbolic notation for mechanism", *Trans. of ASME, J. of Engineering for Industry*, Vol. 93, 1971, p. 102-112.
- [**Shiller 94**] Shiller Z., "On singular time-optimal control along specified paths", *IEEE Trans. on Robotics and Automation*, Vol. RA-10(4), 1994, p. 561-566.
- [**Shin 85**] Shin K.G., McKay N.D., "Minimum time control of robotic manipulators with geometric path constraints", *IEEE Trans. on Automatic Control*, Vol. AC-30(6), 1985, p. 531-541.
- [**Siciliano 93**] Siciliano B., Villani L., "An adaptive force/position regulator for robot manipulators", *Int. J. of Adaptive Control and Signal Processing*, Vol. 7, 1993, p. 389-403.
- [**Siciliano 96a**] Siciliano B., Villani L., "A passivity-based approach to force regulation and motion control of robot manipulators", *Automatica*, Vol. 32, 1996, p. 443-447.
- [**Siciliano 96b**] Siciliano B., Villani L., "Adaptive compliant control of robot manipulators", *Control Engineering Practice*, Vol. 4, 1996, p. 705-712.
- [**Siciliano 00**] Siciliano B., Villani L., *Robot force control*, Kluwer Academic Publishers, Boston, 2000.
- [**Siciliano 2008**] Siciliano B. Khatib O, eds., *Springer hand book of Robotics*, Springer Verlag, 2008.
- [**Slotine 87**] Slotine J.-J.E., Li W., "Adaptive manipulator control: a case study", *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, March-April 1987, p. 1312-1400.
- [**Slotine 91**] Slotine J.-J.E., Li W., *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, 1991.
- [**Smith 73**] Smith D.A., Chace M.A., Rubens A.C., "The automatic generation of a mathematical model for machinery systems", *Trans. of ASME, J. of Engineering for Industry*, 1973, Vol. 95, p. 629-635.
- [**Spanos 85**] Spanos J., Kholi D., "Workspace analysis of regional structures of manipulators", *J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 107, June 1985, p. 219-225.
- [**Spetch 88**] Spetch R., Isermann R., "On-line identification of inertia, friction and gravitational forces applied to an industrial robot", *Proc. IFAC Symp. on Robot Control, SYROCO'88*, 1988, p. 88.1-88.6.

- [Spong 87] Spong M.W., "Modeling and control of elastic joint robots", *Trans. of the ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 109, 1987, p. 310-319.
- [Spong 89] Spong M.W., Vidyasagar M., *Robot dynamics and control*, John Wiley & Sons, New York, 1989.
- [Spong 90] Spong M.W., ???
- [Stepanenko 93] Stepanenko Y., Yuan J., "A reduced order regressor and its application to adaptive robotic control", *The Int. J. of Robotics Research*, Vol. 12(2), April 1993, p. 180-187.
- [Stewart 65] Stewart D., "A platform with six degrees of freedom", *Proc. of Institution of Mechanical Engineers*, Vol. 180, Part 1, n° 15, 1965-1966, p. 371-385.
- [Sugimoto 85] Sugimoto K., Okada T., "Compensation of positioning errors caused by geometric deviations in robot systems", *The 3rd Int. Symp. of Robotics Research*, MIT Press, Cambridge, 1985, p. 231-236.
- [Sugimoto 89] Sugimoto K., "Computational scheme for dynamic analysis of parallel manipulators", *Transaction of the ASME, J. of mechanics, Transmission and Automation in Design*, Vol. 111, p.29-33 (1989).
- [Swevers 97] Swevers J., Ganseman C., Tükel D.B., DE Schutter, J.D., Van Brussel H., "Optimal robot excitation and identification", *IEEE Transactions on Robotics and Automation*, Vol.13(5), 1997, p. 730-740.
- [Takegaki 81a] Takegaki M., Arimoto S., "An adaptive trajectory control of manipulators", *Int. J. Control*, Vol. 34(2), 1981, p. 219-230.
- [Takegaki 81b] Takegaki M., Arimoto S., "A new feedback method for dynamic control of manipulators", *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 102, 1981, p. 119-125.
- [Tancredi 95] Tancredi L., "De la simplification et la résolution du modèle géométrique direct des robots parallèles", Thèse de Doctorat, Ecole Nationale Supérieure des Mines de Paris, France, December 1995.
- [Tang 94] Tang G.R., Lieu L.S., "A study of three robot calibration methods based on flat surfaces", *J. of Mechanism and Machine Theory*, Vol. 29(2), 1994, p. 195-206.
- [Taylor 79] Taylor R.H., "Planning and execution of straight line manipulator trajectories", *IBM J. of Research and Development*, Vol. 23(4), July 1979, p. 424-436.
- [Thérond 96] Thérond X., Dégoulange E., Dombre E., Pierrot F., "Force control of a medical robot for arterial diseases detection", *Proc. 6th Int. Symp. on Robotics and Manufacturing, WAC'96*, Montpellier, May 1996, Vol. 6, p. 793-798.
- [Tomei 91] Tomei P., "Adaptive PD controller for robot manipulators", *IEEE Trans. on Robotics and Automation*, Vol. RA-7(4), 1991, p. 565-570.
- [Tondou 94] Tondou B., El Zorkany H., "Identification of a trajectory model for the PUMA-560 Robot", *J. of Robotic Systems*, Vol. 11(2), 1994, p. 77-90.
- [Touron 84] Touron P., "Modélisation de la dynamique des mécanismes polyarticulés ; application à la CAO et à la simulation de robots", Thèse de Docteur-Ingénieur, USTL, Montpellier, France, July 1984.
- [Tsai 2000] Tsai L-W., "Solving the inverse dynamics of a Stewart-Gough

manipulator by the principle of virtual work”, *J. of Mechanical design* **122**, p. 3-9 (March 2000).

- [**Uicker 69**] Uicker J.J., "Dynamic behavior of spatial linkages", *Trans. of ASME, J. of Engineering for Industry*, Vol. 91, 1969, p. 251-258.
- [**Vandanjon 95**] Vandanjon P.-O., Gautier M., Desbats P., "Identification of inertial parameters of robots by means of spectrum analysis", *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, May 1995, p. 3033-3038.
- [**Veitschegger 86**] Veitschegger W.K., Wu C.H., "Robot accuracy analysis based on kinematics", *IEEE J. of Robotics and Automation*, Vol. RA-2(3), September 1986, p. 171-179.
- [**Venture 2006**] G.Venture, P-J Ripert, W.Khalil, M.Gautier, Ph. Bodson, "Modeling and identification of passenger car dynamics using robotics formalism", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, issue 3, p. 349 – 359, 2006,
- [**Volpe 93**] Volpe R., Khosla P., "A theoretical and experimental investigation of explicit force control for manipulators", *IEEE Trans. on Automatic Control*, Vol. AC-38(11), 1993, p. 1634-1650.
- [**Volpe 95**] Volpe R., Khosla P., "The equivalence of second-order impedance control and proportional gain explicit force control", *The Int. J. of Robotics Research*, Vol. 14(6), 1995, p. 574-589.
- [**Vukobratovic 82**] Vukobratovic M., Potkonjak V., *Dynamics of manipulation robots*; Vol. 1: *Theory and applications*, Springer-Verlag, New York, 1982.
- [**Vukobratovic 85**] Vukobratovic M., Kircanski N., *Real-time dynamics of manipulation robots*, Springer-Verlag, New York, 1985.
- [**Walker 82**] Walker M.W., Orin D.E., "Efficient dynamic computer simulation of robotics mechanism", *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 104, 1982, p. 205-211.
- [**Wampler 86**] Wampler C.W., "Manipulator inverse kinematic solutions based on vector formulation and damped least-squares methods", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-16, 1986, p. 93-101.
- [**Wang 83**] Wang L.T., Ravani B., "Recursive computations of kinematic and dynamic equations for mechanical manipulators", *IEEE J. of Robotics and Automation*, Vol. RA-1(3), September 1983, p. 124-131.
- [**Wang 91**] Wang L.C.T., Chen C.C., "A combined optimisation method for solving the inverse kinematics problem of mechanical manipulators", *IEEE Trans. on Robotics and Automation*, Vol. RA-7(4), 1991, p. 489-499.
- [**Wang 93**] Wang D., McClamroch N.H., "Position and force control for constrained manipulator motion: Lyapunov's direct method", *IEEE Trans. on Robotics and Automation*, Vol. RA-9(3), 1993, p. 308-313.
- [**Wen 88**] Wen J. T., Bayard D., "New class of control laws for robotic manipulators; Part 1: non-adaptive case", *Int. J. Control*, Vol. 47(5), 1988, p. 1361-1385.
- [**Wen 91**] Wen J., Murphy S., "Stability analysis of position and force control for robot arms", *IEEE Trans. on Automatic Control*, Vol. AC-36, 1991, p. 365-371.

- [Wenger 89] Wenger P., "Aptitude d'un robot manipulateur à parcourir son espace de travail en présence d'obstacles", Thèse de Doctorat, ENSM, Nantes, France, September 1989.
- [Wenger 92] Wenger P., "A new general formalism for the kinematic analysis of all non redundant manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, Nice, May 1992, p. 442-447.
- [Wenger 93] Wenger P., "A classification of manipulator geometries based on singularity avoidance ability", *Proc. Int. Conf. on Advanced Robotics, ICAR'93*, Tokyo, November 1993, p. 649-654.
- [West 89] West H., Papadopoulos E., Dubowsky S., Cheah H., "A method for estimating the mass properties of a manipulator by measuring the reaction moments at its base", *Proc. IEEE Int. Conf. on Robotics and Automation*, Scottsdale, May 1989, p. 1510-1516.
- [Whitney 69] Whitney D.E., "Resolved motion rate control of manipulators and human prostheses", *IEEE Trans. on Man Machine Systems*, Vol. MMS-10(2), June 1969, p. 47-53.
- [Whitney 72] Whitney D.E., "The mathematics of coordinated control of prosthetic arms and manipulators", *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 94, December 1972, p. 303-309.
- [Whitney 79] Whitney D.E., Nevins J.L., "What is the remote center compliance (RCC) and what can it do", *Proc. 9th Int. Symp. on Industrial Robots*, Washington, March 1979, p. 135-147.
- [Whitney 85] Whitney D.E., "Historical perspective and state of the art in robot force control", *Proc. IEEE Int. Conf. on Robotics and Automation*, St Louis, March 1985, p. 262-268.
- [Whitney 86] Whitney D.E., Lozinski C.A., Rourke J.M., "Industrial robot forward calibration method and results", *J. Dynamic Systems, Measurements, and Control*, Vol. 108, p. 1-8, 1986.
- [Wittenburg 77] Wittenburg J., *Dynamics of system of rigid bodies*, B.G. Teubner, Stuttgart, 1977.
- [Wittenburg 85] Wittenburg J., Holtz U., "The program MESA VERDE for robot dynamics simulations", *The 3rd Int. Symp. of Robotics Research*, MIT Press, Cambridge, 1985, p. 197-204.
- [Wu 84] Wu C.H., "A kinematic CAD tool for the design and control of robot manipulators", *The Int. J. of Robotics Research*, Vol. 3(1), 1984, p. 74-85.
- [Yabuta 92] Yabuta T., "Nonlinear basic stability concept of the hybrid position/force control scheme for robot manipulators", *IEEE Trans. on Robotics and Automation*, Vol. RA-8(5), 1992, p. 663-670.
- [Yang 66] Yang A.T., Freudenstein F., "Application of dual number quaternion algebra in the analysis of spatial mechanisms", *Trans. of ASME, J. of Applied Mechanics*, Vol. 33, 1966, p. 300-308.
- [Yoshikawa 84a] Yoshikawa T., "Analysis and control of robot manipulators with redundancy", *The 1st Int. Symp. of Robotics Research*, MIT Press, Cambridge, 1984, p. 735-748.

- [**Yoshida 00**] Yoshoda K., Khalil W., "Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters", *Int. Journal of robotics research*, 2000, Vol. 19, No. 5, may 2000, pp. 498-510.
- [**Yoshikawa 84b**] Yoshikawa T., "Manipulability of robotics mechanisms", *Proc. 2nd Int. Symp. of Robotics Research*, Kyoto, August 1984, p. 91-98.
- [**Zabala 78**] Zabala Iturralde J., "Commande des robots-manipulateurs à partir de la modélisation de leur dynamique", Thèse de Troisième Cycle, UPS, Toulouse, France, July 1978.
- [**Zeghloul 91**] Zeghloul S., "Développement d'un système de CAO Robotique intégrant la planification de tâches et la synthèse de sites robotisés", Thèse d'Etat, Université de Poitiers, France, February 1991.
- [**Zgaib 92**] Zghaib W., "Génération symbolique automatique des équations de la dynamique des systèmes mécaniques complexes avec contraintes cinématiques", Thèse de Doctorat, ENSAM, Paris, France, 1992.
- [**Zhang 92**] Zhang C., Song S.M., "Forward kinematics of a class of parallel (Stewart) platforms with closed-form solutions", *J. of Robotic Systems*, Vol. 9(1), 1992, p. 93-112.
- [**Zhong 95**] Zhong X.L., Lewis J.M., "A new method for autonomous robot calibration", *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, May 1995, p. 1790-1795.
- [**Zhong 99**] Zhong X.L., ???
- [**Zhuang 95**] Zhuang H., Masory O., Yan J., "Kinematic calibration of a Stewart platform using pose measurements obtained by a single theodolite", *Proc. 1995 Int. Conf. on Intelligent Robots and Systems, IROS'95*, Pittsburg, August 1995, p. 329-334.
- [**Zhuang 97**] Zhuang H., "Self-calibration of a class of parallel mechanisms with a case study on Stewart platform", *IEEE Trans. on Robotics and Automation*, Vol. RA-13(3), 1997, p. 387-397.
- [**Zodiac 96**] The Zodiac, *Theory of robot control*, C. Canudas de Wit, B. Siciliano, G. Bastin Eds., Springer-Verlag, Berlin, 1996.

Modélisation, identification et commande des robots

Appendix 1

Solution of the inverse geometric model equations (Table 4.1)

A1.1. Type 2

The equation to be solved is:

$$X S\theta_i + Y C\theta_i = Z \quad [A1.1]$$

Four cases are possible:

i) if $X = 0$ and $Y \neq 0$, we can write that:

$$C\theta_i = \frac{Z}{Y} \quad [A1.2]$$

yielding:

$$\theta_i = \text{atan2}(\pm \sqrt{1 - (C\theta_i)^2}, C\theta_i) \quad [A1.3]$$

ii) if $Y = 0$ and $X \neq 0$, we obtain:

$$S\theta_i = \frac{Z}{X} \quad [A1.4]$$

yielding:

$$\theta_i = \text{atan2}(S\theta_i, \pm \sqrt{1 - (S\theta_i)^2}) \quad [A1.5]$$

iii) if X and Y are not zero, and $Z = 0$:

$$\begin{cases} \theta_i = \text{atan2}(-Y, X) \\ \theta_i' = \theta_i + \pi \end{cases} \quad [\text{A1.6}]$$

(if $X = Y = 0$, the robot is in a singular configuration);

iv) if X , Y and Z are not zero, we can write that [Gorla 84]:

$$Y C\theta_i = Z - X S\theta_i \quad [\text{A1.7}]$$

Squaring the equation leads to:

$$Y^2 C^2\theta_i = Y^2 (1 - S^2\theta_i) = Z^2 - 2Z X S\theta_i + X^2 S^2\theta_i \quad [\text{A1.8}]$$

Therefore, we have to solve a second degree equation in $S\theta_i$. Likewise, we can write an equation in $C\theta_i$. Finally, we obtain:

$$\begin{cases} S\theta_i = \frac{XZ + \varepsilon Y \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \\ C\theta_i = \frac{YZ - \varepsilon X \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \end{cases} \quad [\text{A1.9}]$$

with $\varepsilon = \pm 1$ (it is straightforward to verify that two combinations of $S\theta_i$ and $C\theta_i$ can only satisfy the original equation). If $X^2 + Y^2 \leq Z^2$, there is no solution. Otherwise, the solution is given by:

$$\theta_i = \text{atan2}(S\theta_i, C\theta_i) \quad [\text{A1.10}]$$

A1.2. Type 3

The system of equations to be solved is the following:

$$\begin{cases} X_1 S\theta_i + Y_1 C\theta_i = Z_1 \\ X_2 S\theta_i + Y_2 C\theta_i = Z_2 \end{cases} \quad [\text{A1.11}]$$

Multiplying the first equation by Y_2 and the second by Y_1 , under the condition that $X_1 Y_2 - X_2 Y_1 \neq 0$, yields:

$$S\theta_i = \frac{Z_1 Y_2 - Z_2 Y_1}{X_1 Y_2 - X_2 Y_1} \quad [\text{A1.12}]$$

then, multiplying the first equation by X_2 and the second by X_1 , yields:

$$C\theta_i = \frac{Z_2 X_1 - Z_1 X_2}{X_1 Y_2 - X_2 Y_1} \quad [A1.13]$$

Thus:

$$\theta_i = \text{atan2}(S\theta_i, C\theta_i) \quad [A1.14]$$

The condition $X_1 Y_2 - X_2 Y_1 \neq 0$ means that the two equations of [A1.11] are independent. If it is not the case, we solve one of these equations as a type-2 equation.

In the frequent case where Y_1 and X_2 are zero, the system [A1.11] reduces to:

$$\begin{cases} X_1 S\theta_i = Z_1 \\ Y_2 C\theta_i = Z_2 \end{cases} \quad [A1.15]$$

whose solution is straightforward:

$$\theta_i = \text{atan2}\left(\frac{Z_1}{X_1}, \frac{Z_2}{Y_2}\right) \quad [A1.16]$$

A1.3. Type 4

The system of equations to be solved is given by:

$$\begin{cases} X_1 r_j S\theta_i = Y_1 \\ X_2 r_j C\theta_i = Y_2 \end{cases} \quad [A1.17]$$

We first compute r_j by squaring both equations and adding them; then, we obtain θ_i by solving a type-3 system of equations:

$$\begin{cases} r_j = \pm \sqrt{(Y_1/X_1)^2 + (Y_2/X_2)^2} \\ \theta_i = \text{atan2}\left(\frac{Y_1}{X_1 r_j}, \frac{Y_2}{X_2 r_j}\right) \end{cases} \quad [A1.18]$$

A1.4. Type 5

The system of equations to be solved is as follows:

$$\begin{cases} X1 S\theta_i = Y1 + Z1 r_j \\ X2 C\theta_i = Y2 + Z2 r_j \end{cases} \quad [A1.19]$$

Let us normalize the equations such that:

$$\begin{cases} S\theta_i = V1 + W1 r_j \\ C\theta_i = V2 + W2 r_j \end{cases} \quad [A1.20]$$

After squaring both equations and adding them, we obtain a second degree equation in r_j , which can be solved if:

$$[W1^2 + W2^2 - (V1 W2 - V2 W1)^2] > 0 \quad [A1.21]$$

Then, we obtain θ_i by solving a type-3 system of equation.

A1.5. Type 6

The system of equations is given by:

$$\begin{cases} W S\theta_j = X C\theta_i + Y S\theta_i + Z1 \\ W C\theta_j = X S\theta_i - Y C\theta_i + Z2 \end{cases} \quad [A1.22]$$

with $Z1 \neq 0$ and/or $Z2 \neq 0$. By squaring both equations and adding them, we obtain a type-2 equation in θ_j :

$$B1 S\theta_i + B2 C\theta_i = B3 \quad [A1.23]$$

with:

$$\begin{aligned} B1 &= 2 (Z1 Y + Z2 X) \\ B2 &= 2 (Z1 X - Z2 Y) \\ B3 &= W^2 - X^2 - Y^2 - Z1^2 - Z2^2 \end{aligned}$$

Knowing θ_i , we obtain θ_j by solving a type-3 system of equation.

A1.6. Type 7

The system of equations is the following:

$$\begin{cases} W1 C\theta_j + W2 S\theta_j = X C\theta_i + Y S\theta_i + Z1 \\ W1 S\theta_j - W2 C\theta_j = X S\theta_i - Y C\theta_i + Z2 \end{cases} \quad [A1.24]$$

It is a generalized form of a type-6 system. Squaring both equations and adding them gives a type-2 equation in θ_i :

$$B1 S\theta_i + B2 C\theta_i = B3 \quad [A1.25]$$

where $B3 = W1^2 + W2^2 - X^2 - Y^2 - Z1^2 - Z2^2$. The terms $B1$ and $B2$ are identical to those of equation [A1.23].

After solving for θ_i , we compute θ_j as a solution of a type-3 system of equation.

A1.7. Type 8

The system of equations is the following:

$$\begin{cases} X C\theta_i + Y C(\theta_i + \theta_j) = Z1 \\ X S\theta_i + Y S(\theta_i + \theta_j) = Z2 \end{cases} \quad [A1.26]$$

By squaring both equations and adding them, θ_i vanishes, yielding:

$$C\theta_j = \frac{Z1^2 + Z2^2 - X^2 - Y^2}{2XY} \quad [A1.27]$$

hence:

$$\theta_j = \text{atan2}(\pm \sqrt{1 - (C\theta_j)^2}, C\theta_j) \quad [A1.28]$$

Then, [A1.26] reduces to a system of two equations in θ_i such that:

$$\begin{cases} S\theta_i = \frac{B1Z2 - B2Z1}{B1^2 + B2^2} \\ C\theta_i = \frac{B1Z1 + B2Z2}{B1^2 + B2^2} \end{cases} \quad [A1.29]$$

with $B1 = X + Y C\theta_j$ and $B2 = Y S\theta_j$. Finally:

$$\theta_i = \text{atan2}(S\theta_i, C\theta_i) \quad [A1.30]$$

Appendix 2

The inverse robot

The n degree-of-freedom robot whose set of geometric parameters are $(\sigma_j', \alpha_j', d_j', \theta_j', r_j')$ is defined as the inverse of the robot $(\sigma_j, \alpha_j, d_j, \theta_j, r_j)$ if the transformation matrix ${}^0T_n(\sigma_j', \alpha_j', d_j', \theta_j', r_j')$ is equal to ${}^0T_n^{-1}(\sigma_j, \alpha_j, d_j, \theta_j, r_j)$.

Table A2.1 gives the geometric parameters of a general six degree-of-freedom robot. Table A2.2 gives those of the corresponding inverse robot. Indeed, let us write the transformation matrix 0T_6 under the following form:

$${}^0T_6 = \mathbf{Rot}(z, \theta_1) \mathbf{Trans}(z, r_1) \mathbf{Rot}(x, \alpha_2) \mathbf{Trans}(x, d_2) \mathbf{Trans}(z, r_2) \mathbf{Rot}(z, \theta_2) \dots \mathbf{Rot}(x, \alpha_6) \mathbf{Trans}(x, d_6) \mathbf{Trans}(z, r_6) \mathbf{Rot}(z, \theta_6) \quad [A2.1]$$

Table A2.1. *Geometric parameters of a general six degree-of-freedom robot*

j	σ_j	α_j	d_j	θ_j	r_j
1	σ_1	0	0	θ_1	r_1
2	σ_2	α_2	d_2	θ_2	r_2
3	σ_3	α_3	d_3	θ_3	r_3
4	σ_4	α_4	d_4	θ_4	r_4
5	σ_5	α_5	d_5	θ_5	r_5
6	σ_6	α_6	d_6	θ_6	r_6

The inverse transformation matrix 6T_0 can be written as:

$${}^6T_0 = \mathbf{Rot}(z, -\theta_6) \mathbf{Trans}(z, -r_6) \mathbf{Trans}(x, -d_6) \mathbf{Rot}(x, -\alpha_6) \mathbf{Rot}(z, -\theta_5) \mathbf{Trans}(z, -r_5) \dots \mathbf{Trans}(x, -d_2) \mathbf{Rot}(x, -\alpha_2) \mathbf{Rot}(z, -\theta_1) \mathbf{Trans}(z, -r_1) \quad [A2.2]$$

The parameters of Table A2.2 result from comparing equations [A2.1] and [A2.2]. The corresponding elementary transformation matrices are denoted by ${}^jT_j'$ such that:

$${}^0T_6' = {}^0T_1' {}^1T_2' \dots {}^5T_6' = {}^0T_6^{-1} \quad [A2.3]$$

Table A2.2. *Geometric parameters of the six degree-of-freedom inverse robot*

j	σ_j'	α_j'	d_j'	θ_j'	r_j'
1	σ_6	0	0	$-\theta_6$	$-r_6$
2	σ_5	$-\alpha_6$	$-d_6$	$-\theta_5$	$-r_5$
3	σ_4	$-\alpha_5$	$-d_5$	$-\theta_4$	$-r_4$
4	σ_3	$-\alpha_4$	$-d_4$	$-\theta_3$	$-r_3$
5	σ_2	$-\alpha_3$	$-d_3$	$-\theta_2$	$-r_2$
6	σ_1	$-\alpha_2$	$-d_2$	$-\theta_1$	$-r_1$

Appendix 3

Dyalitic elimination

Let us consider the following system of equations in the two unknowns x, y :

$$\begin{cases} a x^2 y^2 + b xy = c y + d \\ e x^2 y^2 + f xy + g = 0 \end{cases} \quad [\text{A3.1}]$$

where the coefficients a, b, \dots, g are constants with arbitrary values. The so-called *dyalitic elimination technique* [Salmon 1885] consists of:

i) transforming the system [A3.1] as a linear system such that:

$$\begin{bmatrix} ax^2 & bx-c & -d \\ ex^2 & fx & g \end{bmatrix} \begin{bmatrix} y^2 \\ y \\ 1 \end{bmatrix} = \mathbf{0} \quad [\text{A3.2}]$$

where y^2, y and 1 are termed *power products*;

ii) *increasing the number of equations*: by multiplying both equations by y , we obtain two new equations that form, together with those of [A3.2], a homogeneous system consisting of four equations in four unknowns (power products):

$$\mathbf{M} \mathbf{Y} = \mathbf{0} \quad [\text{A3.3}]$$

where \mathbf{M} is a function of x :

$$\mathbf{M} = \begin{bmatrix} 0 & ax^2 & bx-c & -d \\ 0 & ex^2 & fx & g \\ ax^2 & bx-c & -d & 0 \\ ex^2 & fx & g & 0 \end{bmatrix} \text{ and } \mathbf{Y} = [y^3 \ y^2 \ y \ 1]^T$$

Since one of the elements of \mathbf{Y} is 1, the system [A3.3] is compatible if, and only if, it is singular, which implies that the determinant of \mathbf{M} is zero. Applying this condition to the example leads to a fourth degree equation in x . For each of the four roots, we obtain a different matrix \mathbf{M} . By choosing three equations out of the system [A3.3], we obtain a system of three linear equations of type $\mathbf{A} \mathbf{Y}' = \mathbf{B}$ where $\mathbf{Y}' = [y^3 \ y^2 \ y]^T$. Doing that, each value of x provides a single value of y .

To summarize, the method requires four steps:

- construct the power product equation in order to minimize the number of unknowns;
- add equations to obtain a homogeneous system;
- from this system, compute a polynomial in a single unknown using the fact that the system is necessarily singular;
- compute the other variables by solving a system of linear equations.

Appendix 4

Solution of systems of linear equations

A4.1. Problem statement

Let us consider the following system of m linear equations in n unknowns:

$$\mathbf{Y} = \mathbf{W} \boldsymbol{\zeta} \quad [\text{A4.1}]$$

where \mathbf{W} is an $(m \times n)$ known matrix, \mathbf{Y} is an $(m \times 1)$ known vector and $\boldsymbol{\zeta}$ is the unknown $(n \times 1)$ vector.

Let \mathbf{W}_a be the augmented matrix defined by:

$$\mathbf{W}_a = [\mathbf{W} : \mathbf{Y}]$$

Let r and r_a denote the ranks of \mathbf{W} and \mathbf{W}_a respectively. The relation between r and r_a can be used to analyze the existence of solutions:

- a) if $r = r_a$, the system has at least one solution:
 - if $r = r_a = n$, there is a unique solution;
 - if $r = r_a < n$, the number of solutions is infinite; the system is redundant. For example, this case is encountered with the inverse kinematic model (Chapter 6).
- b) if $r \neq r_a$, the system [A4.1] is not compatible, meaning that it has no exact solution; it will be written as:

$$\mathbf{Y} = \mathbf{W} \boldsymbol{\zeta} + \boldsymbol{\rho} \quad [\text{A4.2}]$$

where $\boldsymbol{\rho}$ is the residual vector or error vector. This case occurs when identifying the geometric and dynamic parameters (Chapters 11 and 12 respectively) or when solving the inverse kinematic model in the vicinity of singular configurations.

A4.2. Resolution based on the generalized inverse

A4.2.1. Definitions

The matrix $\mathbf{W}^{(-1)}$ is a *generalized inverse* of \mathbf{W} if:

$$\mathbf{W} \mathbf{W}^{(-1)} \mathbf{W} = \mathbf{W} \quad [\text{A4.3}]$$

If \mathbf{W} is square and regular, then $\mathbf{W}^{(-1)} = \mathbf{W}^{-1}$. In addition, $\mathbf{W}^{(-1)}$ is said to be a left inverse or a right inverse respectively if:

$$\mathbf{W}^{(-1)} \mathbf{W} = \mathbf{I} \text{ or } \mathbf{W} \mathbf{W}^{(-1)} = \mathbf{I} \quad [\text{A4.4}]$$

It can be shown that \mathbf{W} has an infinite number of generalized inverses unless it is of dimension $(n \times n)$ and of rank n . A solution of the system [A4.1], when it is compatible, is given by:

$$\boldsymbol{\zeta} = \mathbf{W}^{(-1)} \mathbf{Y} \quad [\text{A4.5}]$$

All the solutions are given by the general equation:

$$\boldsymbol{\zeta} = \mathbf{W}^{(-1)} \mathbf{Y} + (\mathbf{I} - \mathbf{W}^{(-1)} \mathbf{W}) \mathbf{Z} \quad [\text{A4.6}]$$

where \mathbf{Z} is an arbitrary $(n \times 1)$ vector. Note that:

$$\mathbf{W} (\mathbf{I} - \mathbf{W}^{(-1)} \mathbf{W}) \mathbf{Z} = \mathbf{0} \quad [\text{A4.7}]$$

Therefore, the term $(\mathbf{I} - \mathbf{W}^{(-1)} \mathbf{W}) \mathbf{Z}$ is a projection of \mathbf{Z} on the null space of \mathbf{W} .

A4.2.2. Computation of a generalized inverse

The matrix \mathbf{W} is partitioned in the following manner:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22} \end{bmatrix} \quad [\text{A4.8}]$$

where \mathbf{W}_{11} is a regular $(r \times r)$ matrix, and r is the rank of \mathbf{W} . Then, it can be verified that:

$$\mathbf{W}^{(-1)} = \begin{bmatrix} \mathbf{W}_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad [\text{A4.9}]$$

This method gives the solution as a function of r components of \mathbf{Y} . Thus, the accuracy of the result may depend on the selected minor. We will see in the next section that the pseudoinverse method allows us to avoid this limitation.

NOTE.– If the (r,r) matrix \mathbf{W}_{11} built up with the first r rows and the first r columns is not regular, it is always possible to define a matrix \mathbf{W}' such that:

$$\mathbf{W}' = \mathbf{R} \mathbf{W} \mathbf{C} = \begin{bmatrix} \mathbf{W}'_{11} & \mathbf{W}'_{12} \\ \mathbf{W}'_{21} & \mathbf{W}'_{22} \end{bmatrix} \quad [\text{A4.10}]$$

where \mathbf{W}'_{11} is a regular (r,r) matrix. The orthogonal matrices \mathbf{R} and \mathbf{C} permute the rows and columns of \mathbf{W} respectively. The generalized inverse of \mathbf{W} is derived from that of \mathbf{W}' as:

$$\mathbf{W}^{(-1)} = \mathbf{C} (\mathbf{W}')^{(-1)} \mathbf{R} \quad [\text{A4.11}]$$

A4.3. Resolution based on the pseudoinverse

A4.3.1. Definition

The *pseudoinverse* of the matrix \mathbf{W} is the generalized inverse \mathbf{W}^+ that satisfies [Penrose 55]:

$$\begin{cases} \mathbf{W} \mathbf{W}^+ \mathbf{W} = \mathbf{W} \\ \mathbf{W}^+ \mathbf{W} \mathbf{W}^+ = \mathbf{W}^+ \\ (\mathbf{W}^+ \mathbf{W})^T = \mathbf{W}^+ \mathbf{W} \\ (\mathbf{W} \mathbf{W}^+)^T = \mathbf{W} \mathbf{W}^+ \end{cases} \quad [\text{A4.12}]$$

It can be shown that the pseudoinverse always exists and is unique. All the solutions of the system [A4.1] are given by:

$$\boldsymbol{\zeta} = \mathbf{W}^+ \mathbf{Y} + (\mathbf{I} - \mathbf{W}^+ \mathbf{W}) \mathbf{Z} \quad [\text{A4.13}]$$

The first term $\mathbf{W}^+ \mathbf{Y}$ is the solution minimizing the Euclidean norm $\|\boldsymbol{\zeta}\|$. The second term $(\mathbf{I} - \mathbf{W}^+ \mathbf{W}) \mathbf{Z}$, also called *optimization term* or *homogeneous solution*, is the projection of an arbitrary vector \mathbf{Z} of \mathfrak{R}^n on $\mathcal{N}(\mathbf{W})$, the null space of \mathbf{W} , and therefore, does not change the value of \mathbf{Y} . It can be shown that $(\mathbf{I} - \mathbf{W}^+ \mathbf{W})$ is of rank $(n-r)$. Consequently, when the

robot is redundant, this term may be used to optimize additional criteria satisfying the primary task. This property is illustrated by examples in Chapter 6.

When the system [A4.1] is not compatible, it can be shown that the solution $\mathbf{W}^+ \mathbf{Y}$ gives the least-squares solution minimizing the error $\|\mathbf{W} \boldsymbol{\zeta} - \mathbf{Y}\|^2 = \|\mathbf{p}\|^2$.

A4.3.2. Pseudoinverse computation methods

A4.3.2.1. Method requiring explicit computation of the rank [Gorla 84]

Let the matrix \mathbf{W} be partitioned as indicated in equation [A4.8] such that \mathbf{W}_{11} is of full rank r . Using the following notations:

$$\mathbf{W}_1 = \begin{bmatrix} \mathbf{W}_{11} \\ \mathbf{W}_{21} \end{bmatrix} \text{ and } \mathbf{W}_2 = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \end{bmatrix}$$

it can be shown that:

$$\mathbf{W}^+ = \mathbf{W}_2^T (\mathbf{W}_1^T \mathbf{W} \mathbf{W}_2^T)^{-1} \mathbf{W}_1^T \quad [\text{A4.14}]$$

When \mathbf{W} is of full rank, this equation may be simplified as follows:

- if $m > n$: $\mathbf{W} = \mathbf{W}_1 \rightarrow \mathbf{W}^+ = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T$, (\mathbf{W}^+ is then the left inverse of \mathbf{W});
- if $m < n$: $\mathbf{W} = \mathbf{W}_2 \rightarrow \mathbf{W}^+ = \mathbf{W}^T (\mathbf{W} \mathbf{W}^T)^{-1}$, (\mathbf{W}^+ is then the right inverse of \mathbf{W});
- if $m = n$: $\mathbf{W} = \mathbf{W}_1 = \mathbf{W}_2 \rightarrow \mathbf{W}^+ = \mathbf{W}^{-1}$.

If \mathbf{W}_{11} is not of rank r , the orthogonal permutation matrices \mathbf{R} and \mathbf{C} of equation [A4.10] should be used, yielding:

$$\mathbf{W}^+ = \mathbf{C} (\mathbf{W}')^+ \mathbf{R} \quad [\text{A4.15}]$$

A4.3.2.2. Greville method [Greville 60], [Fournier 80]

This recursive algorithm is based on the pseudoinverse properties of a partitioned matrix. It does not require the explicit computation of the rank of \mathbf{W} . Let \mathbf{W} be a partitioned matrix such that:

$$\mathbf{W} = \begin{bmatrix} \mathbf{U} & \mathbf{V} \end{bmatrix} \quad [\text{A4.16}]$$

Its pseudoinverse \mathbf{W}^+ can be written as [Boullion 71]:

$$\mathbf{W}^+ = \begin{bmatrix} \mathbf{U}^+ - \mathbf{U}^+ \mathbf{V} \mathbf{C}^+ - \mathbf{U}^+ \mathbf{V} (\mathbf{I} - \mathbf{C}^+ \mathbf{C}) \mathbf{M} \mathbf{V}^T (\mathbf{U}^+)^T \mathbf{U}^+ (\mathbf{I} - \mathbf{V} \mathbf{C}^+) \\ \mathbf{C}^+ + (\mathbf{I} - \mathbf{C}^+ \mathbf{C}) \mathbf{M} \mathbf{V}^T (\mathbf{U}^+)^T \mathbf{U}^+ (\mathbf{I} - \mathbf{V} \mathbf{C}^T) \end{bmatrix} \quad [\text{A4.17}]$$

with:

$$\begin{aligned} \mathbf{C} &= (\mathbf{I} - \mathbf{U} \mathbf{U}^+) \mathbf{V} \\ \mathbf{M} &= [\mathbf{I} + (\mathbf{I} - \mathbf{C}^+ \mathbf{C}) \mathbf{V}^T (\mathbf{U}^+)^T \mathbf{U}^+ \mathbf{V} (\mathbf{I} - \mathbf{C}^+ \mathbf{C})]^{-1} \end{aligned}$$

If the matrix \mathbf{V} reduces to a single column, a recursive algorithm that does not require any matrix inversion may be employed.

Let \mathbf{W}_k contain the first k columns of \mathbf{W} . If \mathbf{W}_k is partitioned such that the first $(k-1)$ columns are denoted by \mathbf{W}_{k-1} and the k^{th} column is \mathbf{w}_k , then:

$$\mathbf{W}_k = [\mathbf{W}_{k-1} \quad \mathbf{w}_k] \quad [\text{A4.18}]$$

The pseudoinverse \mathbf{W}_k^+ is derived from \mathbf{W}_{k-1}^+ and from the k^{th} column of \mathbf{W} :

$$\mathbf{W}_k^+ = \begin{bmatrix} \mathbf{W}_{k-1}^+ - \mathbf{d}_k \mathbf{b}_k^T \\ \mathbf{b}_k \end{bmatrix} \quad [\text{A4.19}]$$

where:

$$\mathbf{d}_k = \mathbf{W}_{k-1}^+ \mathbf{w}_k \quad [\text{A4.20}]$$

In order to evaluate \mathbf{b}_k , we define:

$$\mathbf{c}_k = \mathbf{w}_k - \mathbf{W}_{k-1} \mathbf{d}_k \quad [\text{A4.21}]$$

then, we compute:

$$\begin{aligned} \mathbf{b}_k = \mathbf{c}_k^+ &= (\mathbf{c}_k^T \mathbf{c}_k)^{-1} \mathbf{c}_k^T && \text{if } \mathbf{c}_k \neq \mathbf{0} \\ &= (1 + \mathbf{d}_k^T \mathbf{d}_k)^{-1} \mathbf{d}_k^T \mathbf{W}_{k-1}^+ && \text{if } \mathbf{c}_k = \mathbf{0} \end{aligned} \quad [\text{A4.22}]$$

This recursive algorithm is initialized by calculating \mathbf{W}_1^+ using equation [A4.14]:

$$\mathbf{W}_1^+ = \mathbf{w}_1^+ = (\mathbf{w}_1^T \mathbf{w}_1)^{-1} \mathbf{w}_1^T \quad (\text{if } \mathbf{w}_1 = \mathbf{0}, \text{ then } \mathbf{W}_1^+ = \mathbf{0}^T). \quad [\text{A4.23}]$$

The pseudoinverse of \mathbf{W} can also be calculated by handling recursively rows instead of columns: physically, it comes to consider the equations sequentially.

• **Example A4.1.** Computation of the pseudoinverse using the Greville method. Let us consider the following matrix:

$$\mathbf{W} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

i) *first iteration (initialization):*

$$\mathbf{W}_1^+ = \begin{bmatrix} 1/5 & 2/5 \end{bmatrix}$$

ii) *second iteration:*

$$\mathbf{d}_2 = 8/5, \mathbf{c}_2 = \begin{bmatrix} 2/5 \\ -1/5 \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} 2 & -1 \end{bmatrix}, \mathbf{W}_2^+ = \begin{bmatrix} -3 & 2 \\ 2 & -1 \end{bmatrix}$$

iii) *third iteration:*

$$\mathbf{d}_3 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \mathbf{c}_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{b}_3 = \begin{bmatrix} 7/6 & -2/3 \end{bmatrix}$$

Finally, the pseudoinverse is:

$$\mathbf{W}^+ = \begin{bmatrix} -11/6 & 4/3 \\ -1/3 & 1/3 \\ 7/6 & -2/3 \end{bmatrix}$$

A4.3.2.3. Method based on the singular value decomposition of \mathbf{W}

The singular value decomposition theory [Lawson 74], [Dongarra 79], [Klema 80] states that for an $(m \times n)$ matrix \mathbf{W} of rank r , there exist orthogonal matrices \mathbf{U} and \mathbf{V} of dimensions $(m \times m)$ and $(n \times n)$ respectively, such that:

$$\mathbf{W} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad [\text{A4.24}]$$

The $(m \times n)$ matrix $\mathbf{\Sigma}$ is diagonal and contains the singular values s_i of \mathbf{W} . They are arranged in a decreasing order such that $s_1 \geq s_2 \geq \dots \geq s_r$. $\mathbf{\Sigma}$ has the following form:

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{S}_{r \times r} & \mathbf{0}_{r \times (n-r)} \\ \mathbf{0}_{(m-r) \times r} & \mathbf{0}_{(m-r) \times (n-r)} \end{bmatrix} \quad [\text{A4.25}]$$

where \mathbf{S} is a diagonal (rxr) matrix of rank r , formed by the non-zero singular values s_i of \mathbf{W} .

The singular values of \mathbf{W} are the square roots of the eigenvalues of the matrices $\mathbf{W}^T \mathbf{W}$ or $\mathbf{W} \mathbf{W}^T$ depending on whether $n < m$ or $n > m$ respectively.

The columns of \mathbf{V} are the eigenvectors of $\mathbf{W}^T \mathbf{W}$ and are called *right singular vectors* or *input singular vectors*. The columns of \mathbf{U} are the eigenvectors of $\mathbf{W} \mathbf{W}^T$ and are called *left singular vectors* or *output singular vectors*.

The pseudoinverse is then written as:

$$\mathbf{W}^+ = \mathbf{V} \mathbf{\Sigma}^+ \mathbf{U}^T \quad [\text{A4.26}]$$

with:

$$\mathbf{\Sigma}^+ = \begin{bmatrix} \mathbf{S}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

This method, known as *Singular Value Decomposition* (SVD) [Maciejewski 89], is often implemented for rank determination and pseudoinverse computation in scientific software packages.

The SVD decomposition of \mathbf{W} makes it possible to evaluate the 2-norm condition number, which can be used to investigate the sensitivity of the linear system to data variations on \mathbf{Y} and \mathbf{W} . Indeed, if \mathbf{W} is a square matrix, and assuming uncertainties $\boldsymbol{\zeta} + \mathbf{d}\boldsymbol{\zeta}$, the system [A4.1] may be written as:

$$\mathbf{Y} + \mathbf{d}\mathbf{Y} = [\mathbf{W} + \mathbf{d}\mathbf{W}] [\boldsymbol{\zeta} + \mathbf{d}\boldsymbol{\zeta}] \quad [\text{A4.27}]$$

The relative error of the solution may be bounded such that:

$$\frac{\|\mathbf{d}\boldsymbol{\zeta}\|_p}{\|\boldsymbol{\zeta}\|_p} \leq \text{cond}_p(\mathbf{W}) \frac{\|\mathbf{d}\mathbf{Y}\|_p}{\|\mathbf{Y}\|_p} \quad [\text{A4.28a}]$$

$$\frac{\|\mathbf{d}\boldsymbol{\zeta}\|_p}{\|\boldsymbol{\zeta} + \mathbf{d}\boldsymbol{\zeta}\|_p} \leq \text{cond}_p(\mathbf{W}) \frac{\|\mathbf{d}\mathbf{W}\|_p}{\|\mathbf{W}\|_p} \quad [\text{A4.28b}]$$

$\text{cond}_p(\mathbf{W})$ is the condition number of \mathbf{W} with respect to the p -norm such that:

$$\text{cond}_p(\mathbf{W}) = \|\mathbf{W}\|_p \|\mathbf{W}^+\|_p \quad [\text{A4.29}]$$

where $\|\cdot\|_p$ denotes a vector p -norm or a matrix p -norm.

The 2-norm condition number of a matrix \mathbf{W} is given by:

$$\text{cond}_2(\mathbf{W}) = \frac{s_{\max}}{s_{\min}} \quad [\text{A4.30}]$$

Notice that the condition number is such that:

$$\text{cond}_2(\mathbf{W}) \geq 1 \quad [\text{A4.31}]$$

NOTES.–

- the p-norm of a vector $\boldsymbol{\zeta}$ is defined by:

$$\|\boldsymbol{\zeta}\|_p = \left(\sum_{i=1}^n |\zeta_i|^p \right)^{1/p} \quad \text{for } p \geq 1 \quad [\text{A4.32}]$$

- the p-norm of a matrix \mathbf{W} is defined by:

$$\|\mathbf{W}\|_p = \max \left\{ \frac{\|\mathbf{W} \boldsymbol{\zeta}\|_p}{\|\boldsymbol{\zeta}\|_p} : \boldsymbol{\zeta} \neq \mathbf{0}_{n,1} \right\} = \max \{ \|\mathbf{W} \boldsymbol{\zeta}\|_p : \|\boldsymbol{\zeta}\|_p = 1 \} \quad [\text{A4.33}]$$

- the 2-norm of a matrix is the largest singular value of \mathbf{W} . It is given by:

$$\|\mathbf{W}\|_2 = s_{\max}$$

- equations similar to [A4.28] can be derived for over determined linear systems.

• **Example A4.2.** Computation of the pseudoinverse with the SVD method. Consider the same matrix as in Example A4.1:

$$\mathbf{W} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

It can be shown that:

$$\mathbf{V} = \begin{bmatrix} 0.338 & 0.848 & -0.408 \\ 0.551 & 0.174 & 0.816 \\ 0.763 & -0.501 & -0.408 \end{bmatrix}, \Sigma = \begin{bmatrix} 6.55 & 0 & 0 \\ 0 & 0.374 & 0 \end{bmatrix}, \mathbf{U} = \begin{bmatrix} 0.57 & -0.822 \\ 0.822 & 0.57 \end{bmatrix}$$

The pseudoinverse is obtained by applying equation [A4.26]:

$$\mathbf{W}^+ = \begin{bmatrix} -1.83 & 1.33 \\ -0.333 & 0.333 \\ 1.17 & -0.667 \end{bmatrix}$$

A4.4. Resolution based on the QR decomposition

Given the system of equations [A4.1], two cases are to be considered depending on whether \mathbf{W} is of full rank or not.

A4.4.1. Full rank system

Let us assume that \mathbf{W} is of full rank. The QR decomposition of \mathbf{W} consists of writing that [Golub 83]:

$$\mathbf{Q}^T \mathbf{W} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0}_{(m-r),n} \end{bmatrix} \quad \text{for } m > n, r = n \quad [\text{A4.34}]$$

$$\mathbf{Q}^T \mathbf{W} = [\mathbf{R} \quad \mathbf{0}_{m,n-r}] \quad \text{for } n > m, r = m \quad [\text{A4.35}]$$

where \mathbf{R} is a regular and upper-triangular ($r \times r$) matrix and where \mathbf{Q} is an orthogonal ($m \times m$) matrix.

For sake of brevity, let us only consider the case $m > n$, which typically occurs when identifying the geometric and dynamic parameters (Chapters 11 and 12 respectively). The case $n > m$ can be similarly handled. The matrix \mathbf{Q} is partitioned as follows:

$$\mathbf{Q} = [\mathbf{Q1} \quad \mathbf{Q2}] \quad [\text{A4.36}]$$

where the dimensions of $\mathbf{Q1}$ and $\mathbf{Q2}$ are ($m \times r$) and $m \times (m-r)$ respectively.

Let us define:

$$\mathbf{G} = \mathbf{Q}^T \mathbf{Y} = \begin{bmatrix} \mathbf{Q1}^T \mathbf{Y} \\ \mathbf{Q2}^T \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{G1} \\ \mathbf{G2} \end{bmatrix} \quad [\text{A4.37}]$$

Since the matrix \mathbf{Q} is orthogonal, it follows that [Golub 83]:

$$\|\mathbf{Y} - \mathbf{W} \boldsymbol{\zeta}\|^2 = \|\mathbf{Q}^T \mathbf{Y} - \mathbf{Q}^T \mathbf{W} \boldsymbol{\zeta}\|^2 = \|\mathbf{G1} - \mathbf{R} \boldsymbol{\zeta}\|^2 + \|\mathbf{G2}\|^2 = \|\rho\|^2 \quad [\text{A4.38}]$$

From equation [A4.38], ζ is the unique solution of the system:

$$\mathbf{R} \zeta = \mathbf{G1} \quad [\text{A4.39}]$$

Since \mathbf{R} is a regular and upper-triangular (rxr) matrix, the system [A4.39] can be easily solved with a backward recursion technique (compute sequentially $\zeta_n, \zeta_{n-1}, \dots$). The norm of the residual for the optimal solution is derived as:

$$\|\rho\|_{\min} = \|\mathbf{G2}\| = \|\mathbf{Q2}^T \mathbf{Y}\| \quad [\text{A4.40}]$$

This solution (when $m > n$ and $r = n$) is identical to that obtained by the pseudoinverse. In order to speed up the computations for systems of high dimensions (for example, this is the case for the identification of the dynamic parameters), we can partition the system [A4.1] into k sub-systems such that:

$$\mathbf{Y}(i) = \mathbf{W}(i) \zeta \quad \text{for } i = 1, \dots, k \quad [\text{A4.41}]$$

Let $\mathbf{Q}(i) = [\mathbf{Q1}(i) \quad \mathbf{Q2}(i)]$ and $\mathbf{R}(i)$ be the matrices obtained after a QR decomposition of the matrix $\mathbf{W}(i)$. The global system reduces to the following system of (nxk) equations in n unknowns:

$$\begin{aligned} \begin{bmatrix} \mathbf{Q1}^T(1)\mathbf{Y}(1) \\ \dots \\ \mathbf{Q1}^T(k)\mathbf{Y}(k) \end{bmatrix} &= \begin{bmatrix} \mathbf{Q1}^T(1)\mathbf{W}(1) \\ \dots \\ \mathbf{Q1}^T(k)\mathbf{W}(k) \end{bmatrix} \zeta \\ \begin{bmatrix} \mathbf{Q1}^T(1)\mathbf{Y}(1) \\ \dots \\ \mathbf{Q1}^T(k)\mathbf{Y}(k) \end{bmatrix} &= \begin{bmatrix} \mathbf{R}(1) \\ \dots \\ \mathbf{R}(k) \end{bmatrix} \zeta \end{aligned} \quad [\text{A4.42}]$$

A4.4.2. Rank deficient system

Again, let us assume that $m > n$ but in this case $r < n$. We permute the columns of \mathbf{W} in such a way that the first columns are independent (the independent columns correspond to the diagonal non-zero elements of the matrix \mathbf{R} obtained after QR decomposition of \mathbf{W}). We proceed by a QR decomposition of the permutation matrix and we obtain :

$$\mathbf{Q}^T \mathbf{W} \mathbf{P} = \begin{bmatrix} \mathbf{R1} & \mathbf{R2} \\ \mathbf{0}_{(m-r),r} & \mathbf{0}_{(m-r),(n-r)} \end{bmatrix} \quad [\text{A4.43}]$$

where \mathbf{P} is a permutation matrix obtained by permuting the columns of an identity matrix, \mathbf{Q} is an orthogonal ($m \times m$) matrix, and $\mathbf{R1}$ is a regular and upper-triangular ($r \times r$) matrix.

Let:

$$\mathbf{P}^T \boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\zeta1} \\ \boldsymbol{\zeta2} \end{bmatrix}$$

From equation [A4.37], we obtain:

$$\begin{aligned} \|\rho\|^2 &= \|\mathbf{Y} - \mathbf{W} \boldsymbol{\zeta}\|^2 = \|\mathbf{Q}^T \mathbf{Y} - \mathbf{Q}^T \mathbf{W} \mathbf{P} \mathbf{P}^T \boldsymbol{\zeta}\|^2 \\ &= \left\| \begin{bmatrix} \mathbf{G1} \\ \mathbf{G2} \end{bmatrix} - \begin{bmatrix} \mathbf{R1} \boldsymbol{\zeta1} + \mathbf{R2} \boldsymbol{\zeta2} \\ \mathbf{0}_{(n-r),1} \end{bmatrix} \right\|^2 \\ &= \|\mathbf{G1} - [\mathbf{R1} \boldsymbol{\zeta1} + \mathbf{R2} \boldsymbol{\zeta2}]\|^2 + \|\mathbf{G2}\|^2 \end{aligned} \quad [\text{A4.44}]$$

$\boldsymbol{\zeta1}$ is the unique solution of the system:

$$\mathbf{R1} \boldsymbol{\zeta1} = \mathbf{G1} - \mathbf{R2} \boldsymbol{\zeta2} \quad [\text{A4.45}]$$

Then, we obtain a family of optimal solutions parameterized by the matrices \mathbf{P} and $\boldsymbol{\zeta2}$:

$$\boldsymbol{\zeta} = \mathbf{P} \begin{bmatrix} \boldsymbol{\zeta1} \\ \boldsymbol{\zeta2} \end{bmatrix} \quad [\text{A4.46}]$$

All solutions provide the minimum norm residual given by equation [A4.40]. We obtain a base solution for $\boldsymbol{\zeta2} = \mathbf{0}_{(n-r),1}$. Recall that the pseudoinverse solution provides the minimum norm residual together with the minimum norm $\|\boldsymbol{\zeta}\|^2$.

