

Chapter 10

How to Manipulate? Kinematics, Dynamics and Architecture of Robot Arms



Bruno Belzile  and David St-Onge 

Learning Objectives

The objective at the end of this chapter is to be able to:

- recognize the architecture and mobilities of a robot arm;
- solve the forward and inverse kinematics problem of serial and parallel manipulators;
- obtain the Jacobian relating the velocities of the joints to the end-effector;
- analyze the Jacobian to obtain the different singularities and understand their physical meaning;
- obtain the equations defining the dynamics of a robotic manipulator.

Introduction

Manipulators are not fundamentally different than any other robotic systems regarding their kinematics and dynamics. They are defined by their number of degrees-of-freedom (DoF) and their architecture, which are critical for the envisioned application. This chapter will provide you with an overview of the kinematics of robot arms, including the direct kinematics problem (DKP), the inverse kinematics problem (IKP) and the different types of singularities and how to find them. As kinematics alone is not sufficient for advanced control, you will need to understand also the dynamics of a robotic manipulator; we will cover it briefly.

B. Belzile (✉) · D. St-Onge
Department of Mechanical Engineering, ÉTS Montréal, 1100 Notre-Dame Street West,
Montreal, QC H3C 1K3, Canada
e-mail: bruno.belzile.1@ens.etsmtl.ca

D. St-Onge
e-mail: david.st-onge@etsmtl.ca

© The Author(s) 2022
D. Herath and D. St-Onge (eds.), *Foundations of Robotics*,
https://doi.org/10.1007/978-981-19-1983-1_10

An Industry Perspective

Juxi Leitner, Co-Founder

LYRO Robotics.



My background is in computer science. I started programming computers when I was young (and there was not much else to do in my very tiny hometown in the middle of the alps).

When I was about 15-16, I realised that most of my code lives in a computer and did not really interact or change things in the real world. I started to become more and more interested in robotics and getting inspired by the movies coming out then, such as *The Matrix*, *I robot*, and *Minority Report* (I wanted to build those spider robots!) So, I looked for ways to learn more about it, and I enrolled in a Joint European Master Degree in Space Robotics.

I have researched robotics in academic settings for over a decade before trying to transfer the technology into real-world applications with our current startup LYRO Robotics.

Initially, I was looking at robot swarms and multi-robot coordination (for space exploration particularly), but I got lucky and was able to attend a summer programme in Lisbon to work with the then in-development iCub European Humanoid. I was fascinated by how easy certain tasks come to us, yet how hard they are for robotic systems, like detecting the world around the robot (even how to decide what to focus your "eyes"/cameras on) or how hard it is to pick up an object, even a simple one from a table in front of the robot.

That was eye-opening, and I got excited by the topic of embodiment and how to integrate perception and smarts with the physicality of the robotic system to enable physical interaction with the world! I still find it fascinating, and it is more than 17 years later :)

Another pivotal moment for me was entering and eventually winning the Amazon Robotics Challenge in 2017. There are specific things that industrial robots were designed for, and it's not picking random objects out of dynamic clutter. Building the team (we were 20+) and designing the robotic system was really

just a lot of fun. The part of solving a real-world problem with fundamental tech we researched for years was particularly exciting (and frustrating at the same time ;)

The win showed that thinking about all the options from hardware to software, is important for designing robots that work. So, we started looking for real-world applications and founded LYRO in 2019 to bring robots to markets that are currently underserved due to various reasons (robots too expensive or too incapable is a big one).

Lot of the theory discussed in this chapter are relevant in the real world. For example, the iCub was inspired by the kinematics of a young child. In particular, the hand has a lot of degrees of freedom, three in the shoulder, two in the elbow, and two in the wrist. Then the hand has nine more (given it has five digits). It highlighted an interesting issue for me that the forward kinematics is pretty straightforward (if you have correct measurements), but inverse kinematics, like when I have a position of an object I want to grasp, how do I need to move my various joints, is a very hard and tricky problem with singularities and non-linearities everywhere.

During my PhD, we regularly had to fix the cables in the iCub's arm due to us running into (or over) limits and breaking things!

I work in Robotic Grasping, and the advent of machine learning 20 years ago, and deep learning ten years ago has clearly had an impact. However, while "grasping is solved" is an often-cited quote, it is still non-trivial to get a robotic arm to pick up any random object in any random configuration and perform some useful task with it.

The area is expanding, which is good, but it lacks reproducibility which is slowing down progress.

On the other hand, it is a very exciting time to enter as the whole field shifts more towards robots that perform tasks in a smart fashion rather than "simply" perform the same action over and over again.

Architectures

The physical embodiment of a robotic manipulator (we will use the term robot loosely for this chapter) is a kinematic chain composed by a set of *rigid bodies*, called *links*, connected in series together by *joints* (formally known as *kinematic pairs*). In other words, a joint constrains the motion between two bodies. There are two types of joints, namely *lower kinematic pairs* (LKP) and *higher kinematic pairs* (HKP). By definition, the former involves "a contact taking place along a surface common to the two bodies" (Angeles, 2014). You most likely encountered already the two most common joints that belong to this category: the revolute (rotation, R) and prismatic

(translation, P) joints. While there are also four other types of LKP, helical (screw, H), cylindrical (C), universal (U) and spherical (S), all of them can be obtained with a combination of revolute and prismatic joints. Therefore, the content of this chapter will nearly exclusively focus on those two types of joints. While most joints commonly used in robots only have a single *degree-of-freedom* (DoF), namely the revolute and prismatic joints mentioned above, other types of joints, such as the spherical and cylindrical joints, exist, with, respectively, three and two DoFs. As it will be seen in the subsection on wrist-partitioned serial manipulators, the last three revolute joints of this type of robot are equivalent to a spherical joint.

The architectures of robotic manipulators can be classified into two main categories: *serial* and *parallel*. The former, more common in the manufacturing industry, consist of manipulators made of simple and open kinematic chains. They are known for their reach and simplicity. The Kinova Gen3 lite, shown in Fig. 10.1, falls into this category with its 6 \underline{R} kinematic chain,¹ i.e., an open loop of six actuated revolute joints in a serial array. The latter, parallel manipulators, are based on complex kinematic chains made of at least one loop. They are known for their structural rigidity, speed and the ability to lift a larger payload with respect to the robot mass. While for the serial manipulator, most actuators need to be moved during the robots' motion, the actuators of a parallel manipulator can all be attached rigidly to the base.

Manipulators can also be classified by their **mobility**, which include their DoFs and the type of motion they can generate. For instance, one of the most important type of robotic manipulators is the Schönflies-motion generators (SMG). These 4-DoF robots, capable of three translations and one rotation about an axis of fixed direction (usually the vertical axis), are commonly called SCARA-like robot, after one of the first and well-known SMG, the Selective-Compliance Assembly Robot Arm (SCARA), a serial robot with one prismatic and three revolute joints (Makino et al., 2007). These manipulators can have a serial or a parallel architecture. Nowadays, most industrial manipulator will have 5–7 DoFs, such as the Kinova Gen3 and Gen3 lite.

Kinematics of Serial Manipulators

Serial manipulators are considered simple kinematic chains, i.e., each link can be coupled via one or two joints, to one or two links. The first link is the *base* and the last link is the *end-effector* (EE), sometimes called tool. In the sequel, we will take a closer look to the direct and inverse kinematics of serial manipulators.

Direct Kinematics

Kinematics are used to describe the motion of a robot without considering the dynamics, namely the forces and the torques causing the motion. Therefore, kinematics

¹ An underline letter representing a joint means it is actuated.

Fig. 10.1 Kinova Gen3 lite, a serial 6-DoF robotic manipulator



problems are geometric problems. First, we consider the direct kinematics (DK), sometimes called forward kinematics (FK), of a serial robot. The DK equations are used to **map the joint variables**, called the *posture* or *configuration* of the robot, **into the position and the orientation of the EE**, namely its *pose*. In the end, you will obtain an explicit system of nonlinear equations to compute $\mathbf{p} = [p_x \ p_y \ p_z]^T$, the three-dimensional vector representing the Cartesian position of the EE, as well as a 3×3 orthogonal orientation (rotation) matrix \mathbf{Q} made of three unit vectors parallel to the X -, Y - and Z -axes of the EE (expressed in the base reference frame). Both \mathbf{p} and \mathbf{Q} can be assembled into a single 4×4 homogeneous matrix, as you will see.

Denavit-Hartenberg Convention

It is impossible to discuss the subject of direct kinematics of serial robots without bringing up the *Denavit-Hartenberg convention*. It is a powerful tool that will help you solve the forward kinematics of a serial manipulator in a systematic way. Since this method was first introduced by Hartenberg and Denavit (Hartenberg and Denavit, 1964), some variations were proposed. Here, we use Paul's notation, also known as

the distal variant (Lipkin, 2008). Each link is numbered from 0 to n , 0 being the base, while n is the n th link, namely the flange of the robot to which the end-effector is attached. The i th joint is defined as the one connecting the $(i - 1)$ th and i th links. While the forward kinematics of a serial robot can be solved without the use of the DH convention (or any other), it simplifies considerably the process and can be easily understood by other engineers familiar with the DH notation. Brace yourself, the following lines cover several definitions and formulas, but the procedure quickly becomes easy to use after trying some examples. For each link, a Cartesian frame is defined. Two such frames are shown in Fig. 10.2. You should note that the (X_i, Y_i, Z_i) axes are rigidly attached to the $(i - 1)$ th link. The following convention is used:

1. Z_i is the axis of the i th kinematic pair/joint.
2. X_i is the common normal between $Z_{(i-1)}$ and Z_i . Contrary to Z_i , which does not have a prescribed direction, X_i is oriented from $Z_{(i-1)}$ toward Z_i . If they intersect, resulting in an undefined direction for X_i , the convention is to use the cross product of unit vectors parallel to $Z_{(i-1)}$ and Z_i : $\mathbf{i}_i = \mathbf{k}_{(i-1)} \times \mathbf{k}_i$. In the case the former and the latter are parallel, X_i is arbitrarily chosen to complete the Cartesian frame, i.e., orthogonal to $Z_{(i-1)}$ and Z_i .
3. with the right-hand rule,² Y_i is defined.

With these frames and their respective axes, four parameters are defined for $i = 1 \dots n$: $\theta_i, \alpha_i, d_i, a_i, i = 1 \dots n$, being respectively the *joint angle*, the *link twist*, the *link offset* and the *link length*. They are defined below:

1. a_i is the distance³ between Z_i and $Z_{(i+1)}$ along $X_{(i+1)}$.
2. d_i is the coordinate,⁴ along Z_i , from the origin of the i th frame to the intersection with $X_{(i+1)}$.
3. α_i is the angle between Z_i and $Z_{(i+1)}$, measured with respect to the positive direction of $X_{(i+1)}$.
4. θ_i is the angle between X_i and $X_{(i+1)}$, measured with respect to the positive direction of Z_i .

An homogeneous transformation matrix, as defined in Chap. 6 (Section 6.4.4), is obtained from these parameters, i.e.,

$$\mathbf{H}_{i-1}^i \equiv \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

² As explained in Chap. 4 the thumb of the right hand points along the direction of the Z -axis; the curl of the fingers while closing the hand represents a motion from the X -axis toward the Y -axis.

³ Always positive by definition.

⁴ Being a signed distance, it can be negative.

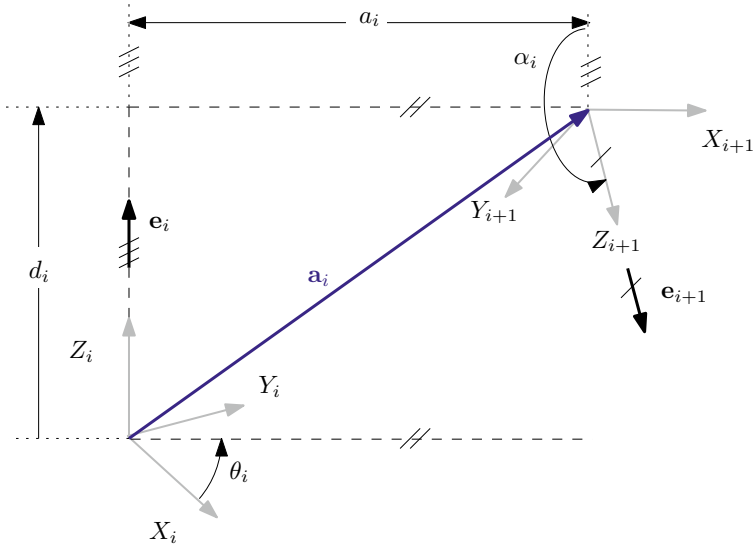


Fig. 10.2 Frames' representation in the DH convention

where subscript refers to the reference frame in which the coordinates are given, $(i - 1)$ in this notation. It can also be separated into the rotation matrix \mathbf{Q}_{i-1}^i and the displacement vector \mathbf{a}_i , i.e.,

$$\mathbf{H}_{i-1}^i = \begin{bmatrix} \mathbf{Q}_{i-1}^i & \mathbf{a}_{i-1}^i \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2)$$

The orientation and position of the EE are thus obtained by multiplying the individual transformation matrices associated with the DH parameters, giving us

$$\mathbf{Q} = \mathbf{Q}_0^1 \mathbf{Q}_1^2 \mathbf{Q}_2^3 \mathbf{Q}_3^4 \mathbf{Q}_4^5 \mathbf{Q}_5^6 \quad (3a)$$

$$\mathbf{p} = \sum_{i=1}^6 \mathbf{a}_0^i \quad \text{or} \quad (3b)$$

$$\mathbf{p} = \mathbf{a}_0^1 + \mathbf{Q}_0^1 \mathbf{a}_1^2 + \mathbf{Q}_0^1 \mathbf{Q}_1^2 \mathbf{a}_2^3 + \mathbf{Q}_0^1 \mathbf{Q}_1^2 \mathbf{Q}_2^3 \mathbf{a}_3^4 + \mathbf{Q}_0^1 \mathbf{Q}_1^2 \mathbf{Q}_2^3 \mathbf{Q}_3^4 \mathbf{a}_4^5 + \mathbf{Q}_0^1 \mathbf{Q}_1^2 \mathbf{Q}_2^3 \mathbf{Q}_3^4 \mathbf{Q}_4^5 \mathbf{a}_5^6 \quad (3c)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{Q} & \mathbf{p} \\ \mathbf{0}^T & 1 \end{bmatrix} = \mathbf{H}_0^1 \mathbf{H}_1^2 \mathbf{H}_2^3 \mathbf{H}_3^4 \mathbf{H}_4^5 \mathbf{H}_5^6 \quad (3d)$$

where \mathbf{H} is the homogeneous transformation matrix representing both the position and orientation of the EE. For the sake of brevity, in the sequel, if only a subscript is given for a rotation/transformation matrix, it is given in the previous reference frame.

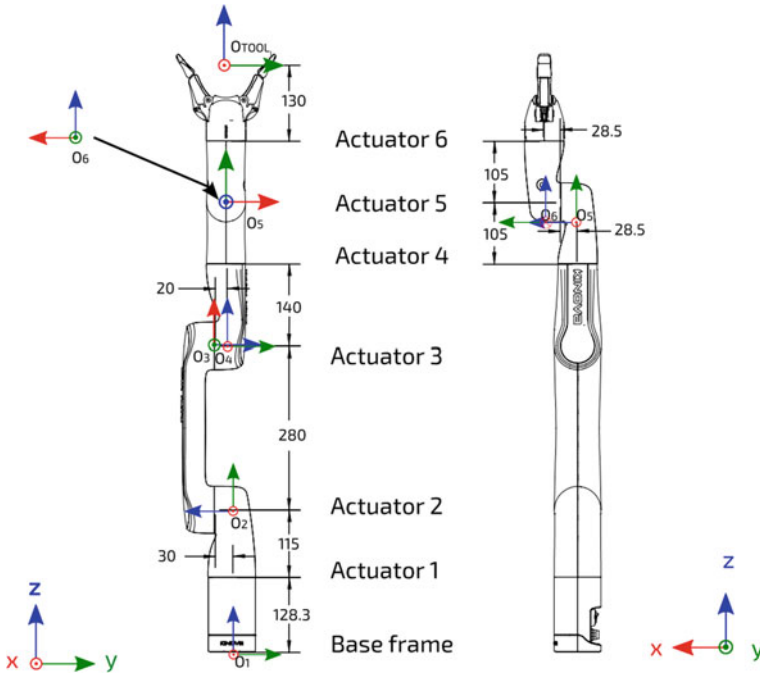


Fig. 10.3 DH frames for each joint for the Kinova Gen3 lite (extracted from the manipulator user manual)

For a joint with a single DoF, such as a revolute or a prismatic joint, only one of the four parameters (a_i , d_i , θ_i , α_i) is a variable, the others are constant. As previously mentioned, a homogeneous transformation matrix is characterized by six parameters in 3D space. Here, this number is reduced to four since, with the DH convention, the location of the origin of frame i is not arbitrarily chosen. Indeed, we have two constraints for the X -axis of each subsequent frame: (1) X_i must be normal to Z_i and (2) it must also intersect it. The frame is rigidly attached to link i , but it is not necessarily located at the end of the link, as one may expect. In fact, it may lie outside the link itself. The reduced number of parameters defining the transformation matrices is one of the main assets of the DH notation.

The DH frames applied to the Kinova Gen3 lite are shown in Fig. 10.3, and the corresponding DH parameters are detailed in Table 10.1. Since the six joints of the Gen3 lite are revolute, all θ_i are unknowns. In Chap. 18, more precisely in Project 3, you will have to find the DH parameters of a 3-DoF version of this manipulator,⁵ as well as compute its forward and inverse kinematics.

⁵ Three of its six joints will be locked.

Table 10.1 DH parameters of the Kinova Gen3 lite

i	1	2	3	4	5	6
a_i	0	a_2	0	0	0	0
d_i	d_1	d_2	d_3	d_4	d_5	d_6
α_i	$\pi/2$	π	$\pi/2$	$\pi/2$	$\pi/2$	0

Inverse Kinematics

As mentioned at the beginning of this chapter, solving the IKP allows the engineer to obtain the set of joint coordinates, namely the posture of the robot, from a position and orientation of the end-effector, namely the pose. Contrary to the DKP, which give only one EE pose from a set of joint coordinates, there may be **more than one solution** to the IKP, i.e., more than one posture that corresponds to a position/orientation of the EE. However, an analytical (symbolic and exact) solution to the inverse kinematics is not necessarily always obtainable, depending on the architecture of the robotic manipulator. In some cases, a numerical approach is preferable. Numerical approaches are also better fit for simulator compatible with various manipulator architectures. The different solutions to the IKP are called configuration types. Usually, while moving, a manipulator will keep the same configuration type, as alternating from one configuration type to another requires large joint angle variations to obtain, in the end, the same EE coordinates. Switching configuration can also risk passing through a singularity, which we will discuss later. The controller of commercially available manipulators takes these elements into account while computing the positions and velocities of the joints.

To solve the IKP symbolically for the explicit equations, we start with the same equations used above, i.e., the ones defined by the 4×4 homogeneous transformation matrix, i.e., \mathbf{H} . Since the last line is always $[0 \ 0 \ 0 \ 1]$, we thus have 12 nonlinear equations, but only six unknowns in the case of a non-redundant⁶ spatial manipulator. Of course, if the robot has additional joints, for example, to reach a target within a cluttered workspace (ex. welding operations), the number of potential solutions increases. Within this chapter, only non-redundant manipulators are considered.

As previously mentioned, while we have nine equations for the orientation of the EE, only three are independent, giving us a system of six equations with six unknowns (three for the orientation, three for the position). Solving the IKP for a general serial manipulator is thus a challenging mathematical problem considering the nonlinearity of the equations. However, you will find that most commercially

⁶ A spatial serial redundant manipulator has more than six joints. Notwithstanding the mechanical limits of the joints, the limits of the reachable workspace and singularities, only six joints are needed to reach any point with any orientation of the EE. You should be careful if you come across the term “redundant,” as it can have different meanings depending on the context. A parallel robot can be redundantly actuated, i.e., more actuators than DoFs, and any manipulator can be *kinematically redundant* with respect to its task, for example, pointing tasks, which only require two DoFs.

available manipulators fall in the special category of *wrist-partitioned*, greatly simplifying the problem, as we will show below.

Wrist-Partitioned Manipulators

The architecture of decoupled serial manipulator (wrist-partitioned) makes it possible to separate the orientation problem from the position problem. Therefore, we obtain *explicit* equations, avoiding the need for a numerical method to solve the IKP. The problem is thus split into the *inverse position kinematics* and the *inverse orientation kinematics*. By definition, the axes of the last three joints of decoupled manipulators intersect. This point is known as the *wrist center*. Looking back to the DH parameters, this means that $a_4 = a_5 = a_6 = 0$. This also means that the last three DH frames share the same origin. The coordinates of the latter are given by vector \mathbf{p}_w in frame 0, i.e.,

$$\mathbf{p}_w = \mathbf{a}_1 + \mathbf{Q}_1 \mathbf{a}_2 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{a}_3 + \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{a}_4 \quad (4)$$

Since $a_4 = 0$, \mathbf{a}_4 is not a function of θ_4 , as the equation of \mathbf{p}_w above. With Eq. (3b), we can rewrite the above equation as

$$\mathbf{p}_w = \mathbf{p} - \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{a}_5 - \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{Q}_4 \mathbf{Q}_5 \mathbf{a}_6 \quad (5)$$

which can be simplify, knowing that with a decoupled wrist, $\mathbf{a}_5 = \mathbf{0}$, as

$$\mathbf{p}_w = \mathbf{p} - \mathbf{Q} \mathbf{Q}_6^T \mathbf{a}_6 \quad (6)$$

This equation is solely function of constant DH parameters and the target position and orientation coordinates of the EE in the case of an IKP. Therefore, the location of the wrist, \mathbf{p}_w , can be computed in the base frame without the joint coordinates, decoupling the position from the orientation.

In short, we solve the position problem by first computing the location of the wrist with Eq. (6), then by isolating the first three joint coordinates in Eq. (4), which is a simpler 3-DoF problem with three equations and three unknowns.

Example: 3-DoF Serial Manipulator

As an example, we can solve 3-DoF inverse position problem for a generic serial manipulator with three revolute joints. It should be noted that the procedure below may need to be slightly adapted in certain special cases (null DH parameters, certain angles, division by zero, etc.). First, we need to rewrite Eq. (4):

$$\mathbf{Q}_1^T (\mathbf{p}_w - \mathbf{a}_1) = \mathbf{a}_2 + \mathbf{Q}_2 \mathbf{a}_3 + \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{a}_4 \quad (7)$$

This can be done because rotation matrices are orthogonal, thus $\mathbf{Q}_i^{-1} = \mathbf{Q}_i^T$. Developing the above equation in terms of its components, we have

$$A \cos \theta_2 + B \sin \theta_2 = x_w \cos \theta_1 + y_w \sin \theta_1 - a_1 \quad (8a)$$

$$A \sin \theta_2 - B \cos \theta_2 = \cos \alpha_1 (y_w \cos \theta_1 - x_w \sin \theta_1) + (z_w - b_1) \sin \alpha_1 \quad (8b)$$

$$C = \sin \alpha_1 (x_w \sin \theta_1 - y_w \cos \theta_1) + (z_w - b_1) \cos \alpha_1 \quad (8c)$$

with

$$A = a_2 + a_3 \cos \theta_3 + b_4 \sin \alpha_3 \sin \theta_3 \quad (8d)$$

$$B = -a_3 \cos \alpha_2 \sin \theta_3 + b_3 \sin \alpha_2 + b_4 \cos \alpha_2 \sin \alpha_3 \cos \theta_3 + b_4 \sin \alpha_2 \cos \alpha_3 \quad (8e)$$

$$C = b_2 + a_3 \sin \alpha_2 \sin \theta_3 + b_3 \cos \alpha_2 - b_4 \sin \alpha_2 \sin \alpha_3 \cos \theta_3 + b_4 \cos \alpha_2 \cos \alpha_3 \quad (8f)$$

We can see that the right-hand side of Eqs (8a–8c) is only function of θ_1 , the position of the wrist and the DH parameters. Let

$$D = x_w \cos \theta_1 + y_w \sin \theta_1 - a_1 \quad (9)$$

$$E = \cos \alpha_1 (y_w \cos \theta_1 - x_w \sin \theta_1) + (z_w - b_1) \sin \alpha_1 \quad (10)$$

we can cast Eq. (8a–8b) in matrix form, i.e.,

$$\begin{bmatrix} A & -B \\ -B & A \end{bmatrix} \begin{bmatrix} \cos \theta_2 \\ \sin \theta_2 \end{bmatrix} = \begin{bmatrix} D \\ E \end{bmatrix} \quad (11)$$

We are now able to compute explicit functions of $\sin \theta_2$ and $\cos \theta_2$:

$$\cos \theta_2 = (AD - BE)/(A^2 + B^2) \quad (12)$$

$$\sin \theta_2 = (BD - AE)/(A^2 + B^2) \quad (13)$$

which leads to

$$\theta_2 = \arctan2(\sin \theta_2, \cos \theta_2) \quad (14)$$

Obviously, θ_2 cannot be computed right away since the values of the other two joint angles are needed. To this aim, we need to make θ_2 disappear. This is done by calculating the sum of squares of each side of Eq. (8a–8c). Knowing $\sin^2 \theta_2 + \cos^2 \theta_2 = 1$, we obtain

$$A^2 + B^2 + C^2 = x_w^2 + y_w^2 + (z_w - b_1)^2 + a_1^2 - 2a_1x_w \cos \theta_1 - 2a_1y_w \sin \theta_1 \quad (15)$$

The left-hand side of the above equation is only a function of DH parameters and θ_3 , while the right-hand side is only dependent on DH parameters and θ_1 . Moreover, Eq. (15) is linear in $\sin \theta_1$, $\sin \theta_3$, $\cos \theta_1$ and $\cos \theta_3$. Computing the sum of the squares of Eq. (8a–8b) would not have been useful, here, to eliminate θ_2 , as the resulting equation would not have been linear in the terms mentioned above, which is necessary for the following steps. Therefore, Eq. (15) is rewritten as

$$F_1 \cos \theta_1 + G_1 \sin \theta_1 + H_1 \cos \theta_3 + I_1 \sin \theta_3 + J_1 = 0 \quad (16)$$

where F_1 , G_1 , H_1 , I_1 and J_1 are only functions of DH parameters and the position of the wrist, all these terms being known at this stage. Then, Eq. (8c) is rewritten in a

similar form, i.e.,

$$F_2 \cos \theta_1 + G_2 \sin \theta_1 + H_2 \cos \theta_3 + I_2 \sin \theta_3 + J_2 = 0 \quad (17)$$

Again, F_2, G_2, H_2, I_2 and J_2 are only functions of DH parameters and the position of the wrist. Having two linear equations and four unknowns, the next step is obtaining explicit expressions of $\cos \theta_1$ and $\sin \theta_1$, as we did with θ_2 . Thus, we obtain

$$\cos \theta_1 = \frac{-G_2(H_1 \cos \theta_3 + I_1 \sin \theta_3 + J_1) + G_1(H_2 \cos \theta_3 + I_2 \sin \theta_3 + J_2)}{F_1 G_2 - F_2 G_1} \quad (18)$$

$$\sin \theta_1 = \frac{F_2(H_1 \cos \theta_3 + I_1 \sin \theta_3 + J_1) - F_1(H_2 \cos \theta_3 + I_2 \sin \theta_3 + J_2)}{F_1 G_2 - F_2 G_1} \quad (19)$$

$$\theta_1 = \arctan2(\sin \theta_1, \cos \theta_1) \quad (20)$$

Finally, we eliminate θ_1 by computing the sum of the $\sin^2 \theta_1$ and $\cos^2 \theta_1$, which results in

$$K \cos^2 \theta_3 + L \sin^2 \theta_3 + M \cos \theta_3 \sin \theta_3 + N \cos \theta_3 + P \sin \theta_3 + Q = 0 \quad (21)$$

where the coefficients in front of the trigonometric functions of θ_3 are functions of F_i, G_i, H_i, I_i and J_i , for $i = 1, 2$, which are in turn functions of DH parameters and the position of the wrist. We, therefore, have a nonlinear equation with known coefficients where the only unknown is θ_3 . To solve this implicit equation, we use a well-known identity in the field of kinematics, the Weierstrass substitution (also known as the tangent half-angle substitution):

$$\cos \theta_3 \equiv \frac{1 - T_3^2}{1 + T_3^2}, \quad \sin \theta_3 \equiv \frac{2T_3}{1 + T_3^2}, \quad T_3 = \tan(\theta_3/2) \quad (22)$$

With this substitution, Eq. (21) is rewritten as an equation of degree four in T_3 :

$$RT_3^4 + ST_3^4 + UT_3^2 + VT_3 + W = 0 \quad (23)$$

All four possible values for T_3 are thus obtained by computing the roots of the above equation. These values are then used to calculate the solutions for θ_3 with

$$\theta_3 = 2 \arctan T_3 \quad (24)$$

The values for the remaining joint coordinates are then computed with first Eq. (20) then Eq. (14), for θ_1 and θ_2 , respectively. Therefore, we have solve the inverse position problem for a 3-DoF serial manipulator, obtaining four sets of joint coordinates. If we replace the revolute joints with prismatic joints, the problem becomes less challenging to solve, as two prismatic joints (and one revolute) lead to a maximum of two solutions to the inverse position problem and three prismatic joints lead to

only one solution to the inverse position problem. The position of the wrist now known; the next step is to find the solutions for the remaining three joints.

Spherical Wrist

The first three rotation matrices \mathbf{Q}_1 , \mathbf{Q}_2 , \mathbf{Q}_3 now fully known; the next step is to compute the solutions for the last three transformation matrices, which are function of the last three joint coordinates. First, we recall Eq. (3a) and rewrite it with everything known on the right, i.e.,

$$\mathbf{Q}_4\mathbf{Q}_5\mathbf{Q}_6 = \mathbf{R} \quad (25a)$$

$$\mathbf{R} = \mathbf{Q}_3^T \mathbf{Q}_2^T \mathbf{Q}_1^T \mathbf{Q}_d = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (25b)$$

Now you should remember that according to the DH notation, the angle between the axes Z_5 and Z_6 is α_5 . These two axes are defined by the unit vectors \mathbf{e}_5 and \mathbf{e}_6 . Therefore, according to the dot product, we have

$$\mathbf{e}_5 \cdot \mathbf{e}_6 = \cos \alpha_5 \quad (26)$$

We need to express these two vectors in one single reference frame. The DH frame 4 is chosen since it simplifies the equations. In this frame, \mathbf{e}_5 is simply the last column of \mathbf{Q}_4 . As for \mathbf{e}_6 is the last column of $\mathbf{Q}_4\mathbf{Q}_5$. To avoid introducing more than one unknown in the equation, we use the fact that

$$\mathbf{Q}_4\mathbf{Q}_5 = \mathbf{R}\mathbf{Q}_6^T \quad (27)$$

We thus obtain an equation where θ_4 is the only unknown variable:

$$X \cos \theta_4 + Y \sin \theta_4 = Z \quad (28a)$$

where

$$X = -\sin \alpha_4 (r_{22} \sin \alpha_6 + r_{23} \cos \alpha_6) \quad (28b)$$

$$Y = \sin \alpha_4 (r_{12} \sin \alpha_6 + r_{13} \cos \alpha_6) \quad (28c)$$

$$Z = -\cos \alpha_4 (r_{32} \sin \alpha_6 + r_{33} \cos \alpha_6) + \cos \alpha_5 \quad (28d)$$

Using the Weierstrass substitution introduced previously, the above equation is then transformed into a quadratic equation in T_4 , where the roots are computed and substituted in $\theta_4 = 2 \arctan T_4$. To find the possible values for the remaining joint angles, we need to go back to Eq. (25a). We keep only the unknown terms on the lefthand side by premultiplying by \mathbf{Q}_4^T , resulting in

$$\mathbf{Q}_5\mathbf{Q}_6 = \mathbf{Q}_4^T \mathbf{R} \quad (29)$$

By developing the components of the above equation and by simple inspection, we find

$$\cos \theta_6 = \frac{r_{12} \sin \alpha_4 \sin \theta_4 - r_{22} \sin \alpha_4 \cos \theta_4 + r_{32} \cos \alpha_4 - \cos \alpha_5 \sin \alpha_6}{\sin \alpha_5 \cos \alpha_6} \quad (30a)$$

$$\sin \theta_6 = \frac{r_{11} \sin \alpha_4 \sin \theta_4 - r_{21} \sin \alpha_4 \cos \theta_4 + r_{31} \cos \alpha_4}{\sin \alpha_5} \quad (30b)$$

As previously done, we put both values into

$$\theta_6 = \arctan2(\sin \theta_6, \cos \theta_6) \quad (31)$$

Finally, θ_5 is found in a similar fashion but with Eq. (27) instead. By inspection, we find

$$\cos \theta_5 = \frac{\cos \alpha_4 \cos \alpha_5 - r_{32} \sin \alpha_6 - r_{33} \cos \alpha_6}{\sin \alpha_4 \sin \alpha_5} \quad (32a)$$

$$\sin \theta_5 = \frac{r_{31} \cos \theta_6 - r_{32} \cos \alpha_6 \sin \theta_6 + r_{33} \sin \alpha_6 \cos \theta_6}{\sin \alpha_4} \quad (32b)$$

and we compute

$$\theta_5 = \arctan2(\sin \theta_5, \cos \theta_5) \quad (33)$$

Other Manipulators

In the case of a serial manipulator without a decoupled wrist, there is no simple recipe to solve the IKP. In some case, a numerical solver is necessary to obtain the joint coordinates from a set of EE coordinates. In other cases, explicit equations can be obtained, for instance, the Kinova Gen3 lite, but they are unique to the robots with the same architecture. However, while the solutions are different, the approach to solve the IKP of non-wrist-partitioned manipulators is generally similar, which is reducing the number of unknowns to only one to obtain the roots of a univariate polynomial equation to compute the values for one joint coordinate, then computing those for the other joints by backsubstitution, as we did with the inverse position problem of the wrist-partitioned manipulator. Indeed, this approach relies mostly on trigonometric identities, e.g.:

- $\sin^2 \theta + \cos^2 \theta = 1$
- $\sin \alpha \sin \beta + \cos \alpha \cos \beta = \cos(\alpha - \beta)$
- $\cos \alpha \cos \beta - \sin \alpha \sin \beta = \cos(\alpha + \beta)$
- $\sin \alpha \cos \beta + \cos \alpha \sin \beta = \sin(\alpha + \beta)$
- $\sin \alpha \cos \beta - \cos \alpha \sin \beta = \sin(\alpha - \beta)$

and the concept of *dyalitic elimination*. The latter is used to reduce the number of unknowns in a system of non-homogeneous equations. The procedure consists of four steps:

1. Rewrite the equations as polynomial expressions where one of the variables is included into the coefficients; this variable is dubbed the *eliminated variable*.
2. As many equations as the number of unknowns is needed; therefore, we may need to generate a new one by multiplying one of the equations by one of the unknowns, for instance, the equations are then casted into matrix form $\mathbf{Ax} = \mathbf{0}$, where \mathbf{A} is a function of powers of the eliminated variable only, and \mathbf{x} of the other unknowns; it should be noted that the last component of \mathbf{x} is equal to 1.
3. Since one component of \mathbf{x} is not equal to zero by definition, \mathbf{A} must be singular; thus, its determinant has to be equal to zero; the next step is thus to compute the roots of $\det(\mathbf{A}) = 0$ to find the possible values of the eliminated variable.
4. The last step is to compute the null space of \mathbf{A} ; knowing the last component must be equal to 1, we simply need to scale the obtain vector to make sure its last component is equal to 1.

Example: IKP of the Kinova Gen3 lite

The inverse kinematics problem of the Kinova Gen3 lite can be solved without the use of a numerical approach. Considering the number of joints, a large set of solutions are obtained for each feasible position and orientation of the end-effector. The methodology to solve the IKP of the Kinova Jaco manipulator, which shares an architecture similar to the Gen3 Lite, can be found in the literature (Gosselin and Liu, 2014). The feasible solutions for an arbitrarily chosen pose of the EE are shown in Fig. 10.4. Four are shown here, but more solutions could have been obtained if we did not take into account the joint rotational limitations. One unique solution can be chosen with a particular criterion, for instance, to minimize the joint rotations, to minimize the torque generated by joint actuator to lift a payload, to simply avoid obstacle, etc. While the topic of the optimal solution to the IKP will not be covered in this chapter, numerous criteria can be found in the literature.

Numerical Approach to the IKP

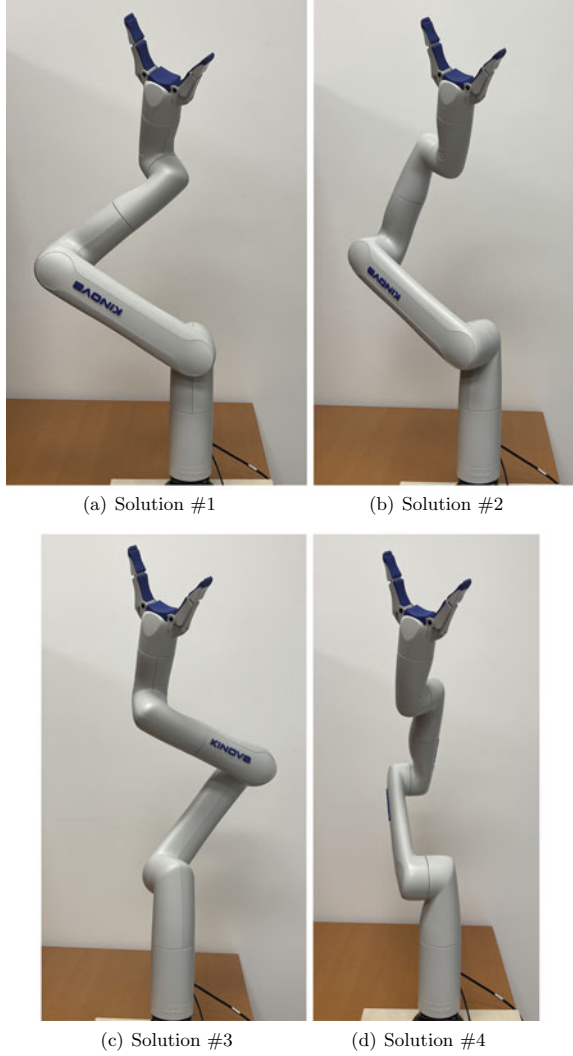
The method presented above to find the symbolic solution to the IKP is not necessarily adequate to all practical use cases. For instance, computing the roots of a high-degree polynomial, which is often the case with manipulators with several DoFs, may lead to numerical instabilities; thus, imprecision on the values of the joint coordinates obtained. The analytical approach may not be fast enough as well. Therefore, to avoid numerical instabilities and finding the symbolic solution to a challenging IKP, the numerical approach is often used in the industry. To this regards, we introduce the Newton-Gauss algorithm, but other avenues are possible. You first need to use the orientation and position of the end-effector to obtain a system of nonlinear equations that can be written as

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \quad (34)$$

Let the desired orientation matrix (defined for instance by Euler angles) and desired position vector

$$\mathbf{Q}_d = [\mathbf{q}_{d,1} \ \mathbf{q}_{d,2} \ \mathbf{q}_{d,3}], \quad \mathbf{p}_d = [p_{x,d} \ p_{y,d} \ p_{z,d}]^T \quad (35)$$

Fig. 10.4 Possible postures for the same EE pose



and the solution to the forward kinematics defined in Eq. (3a–3b). The former can also be shown in a format similar to \mathbf{Q}_d and \mathbf{p}_d :

$$\mathbf{Q} = [\mathbf{q}_1 \ \mathbf{q}_2 \ \mathbf{q}_3], \quad \mathbf{p} = [p_x \ p_y \ p_z]^T \quad (36)$$

For a generic 6-DoF serial manipulator, we thus have a system of 12 equations:

$$\mathbf{f} = \begin{bmatrix} \mathbf{q}_1 - \mathbf{q}_{d,1} \\ \mathbf{q}_2 - \mathbf{q}_{d,2} \\ \mathbf{q}_3 - \mathbf{q}_{d,3} \\ \mathbf{p} - \mathbf{p}_d \end{bmatrix} = \mathbf{0} \quad (37)$$

where \mathbf{f} is a 12-dimensional vector, $\mathbf{0}$ is the null vector of the same dimension and the six unknowns are the joint coordinates we are looking for. The Newton-Gauss algorithm can now be applied to find \mathbf{x} . Through this process, we will find a sequence of approximations of \mathbf{x} , denoted $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ converging toward the solution of the IKP. The next estimation is denoted \mathbf{x}_{k+1} . This algorithm is based on the Taylor series of the first degree; therefore, we have

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k \quad (38a)$$

and

$$\mathbf{f}(\mathbf{x}_{k+1}) = \mathbf{f}(\mathbf{x}_k + \Delta \mathbf{x}_k) = \mathbf{f}(\mathbf{x}_k) + \mathbf{J}_f(\mathbf{x}_k) \Delta \mathbf{x}_k = \mathbf{0} \quad (38b)$$

where $\mathbf{J}_f(\mathbf{x}_k)$ is the mathematical Jacobian of \mathbf{f} with respect to \mathbf{x} (Section 6.6.2), i.e., $\mathbf{J}_f = \partial \mathbf{f} / \partial \mathbf{x}$, evaluated at \mathbf{x}_k . It should not be confused with the Jacobian(s) of the manipulator, which will be introduced later in this chapter. Equation (38b) can be rewritten as

$$\mathbf{J}_f(\mathbf{x}_k) \Delta \mathbf{x}_k = -\mathbf{f}(\mathbf{x}_k) \quad (39)$$

To be able to compute the next increment $\Delta \mathbf{x}_k$ to obtain $\Delta \mathbf{x}_{k+1}$, we thus need to solve the overdetermined system of equation defined by the above equation (\mathbf{J}_f being a tall matrix, i.e., more rows than columns). Since you nearly never have an exact solution for an overdetermined system, we will find the solution minimizing the least squares of the error, known as the least square solution. This is done with the left Moore-Penrose generalized inverse \mathbf{J}_f^L (Section 6.3.3), i.e.,

$$\mathbf{J}_f^L = (\mathbf{J}_f^T \mathbf{J}_f)^{-1} \mathbf{J}_f^T \quad (40a)$$

$$\Delta \mathbf{x}_k = -\mathbf{J}_f^L(\mathbf{x}_k) \mathbf{f}(\mathbf{x}_k) \quad (40b)$$

You should **not** compute the generalized inverse *per se* with the equation above, since it is known to generate numerical issues (the condition number of $\mathbf{J}_f^T \mathbf{J}_f$ is, roughly, the square of that of matrix \mathbf{J}_f itself, resulting into a badly conditioned system (Forsythe, 1970)). Instead, algorithms such as the QR decomposition and the householder reflections are used, achieving the same results while minimizing potential numerical issues.⁷ Depending on the value of \mathbf{x}_1 , the algorithm will converge toward one feasible solution (if any). To obtain at least some of the other potential solutions (thus different configuration types), several starting points \mathbf{x}_1 must be tested.

⁷ Section 6.10.

Jacobian

The forward and inverse kinematics derived in the previous sections relate the joints coordinates to the position and orientation of the end-effector and vice-versa. Now, we consider the velocity of the EE and the joint rates. Mathematically, the relationship between both is the *Jacobian* of the function defining the FKP. The Jacobian is useful to plan smooth trajectory, to compute the *wrench* applied by the EE, to determine singular postures, etc. For your understanding, a wrench is the six-dimensional vector representation of forces and moments. Similarly, a *twist* is the six-dimensional vector representation of linear and angular velocities. The expressions of the twist and the wrench are, respectively,

$$\mathbf{t} \equiv \begin{bmatrix} \boldsymbol{\omega} \\ \dot{\mathbf{p}} \end{bmatrix}, \quad \mathbf{w} \equiv \begin{bmatrix} \mathbf{n} \\ \mathbf{f} \end{bmatrix} \quad (41)$$

where $\dot{\mathbf{p}}$, $\boldsymbol{\omega}$, \mathbf{f} and \mathbf{n} are the 3-dimensional linear velocity, angular velocity, force and moment, respectively.

The Jacobian for a n -link serial manipulator is a $(6 \times n)$ matrix mapping the n joint velocities into the six-dimensional vector consisting of the linear and angular velocities of the EE, i.e., the twist mentioned above. Let us assume only revolute joints for now. Given the angular velocity vector of each link

$$\boldsymbol{\omega}_0 = \mathbf{0} \quad (42a)$$

$$\boldsymbol{\omega}_1 = \dot{q}_1 \mathbf{e}_1 \quad (42b)$$

$$\boldsymbol{\omega}_2 = \dot{q}_2 \mathbf{e}_2 + \boldsymbol{\omega}_1 \quad (42c)$$

$$\boldsymbol{\omega}_3 = \dot{q}_3 \mathbf{e}_3 + \boldsymbol{\omega}_2 \quad (42d)$$

$$\dots$$

$$\boldsymbol{\omega}_n = \dot{q}_n \mathbf{e}_n + \boldsymbol{\omega}_{(n-1)} \quad (42e)$$

where \dot{q}_i is the velocity of the i th joint, \mathbf{e}_i is a unit vector parallel to the axis of the i th joint, namely the Z_i -axis of the i th DH frame, and $\mathbf{0}$ is the three-dimensional null vector. The angular velocity of the end-effector, $\boldsymbol{\omega}$, is simply equal to $\boldsymbol{\omega}_n$. As previously mentioned, the position of the EE is

$$\mathbf{p} = \sum_{i=1}^n \mathbf{a}_i \quad (43)$$

Differentiating the above equation with respect to time, we obtain

$$\dot{\mathbf{p}} = \sum_{i=1}^n \dot{\mathbf{a}}_i, \text{ where } \dot{\mathbf{a}}_i = \boldsymbol{\omega}_i \times \mathbf{a}_i, \quad i = 1, \dots, n \quad (44)$$

Substituting Eqs. (42a) into (44), and with some manipulation, we obtain

$$\dot{\mathbf{p}} = \sum_{i=1}^n \dot{q}_i \mathbf{e}_i \times \mathbf{r}_i, \quad \mathbf{r}_i \equiv \sum_{j=i}^n \mathbf{a}_j \quad (45)$$

where \mathbf{r}_i is defined as the vector from the i th DH frame to the last DH frame attached to the EE. We can rewrite the previous equations in a more compact matrix form:

$$\boldsymbol{\omega} = \mathbf{A}\dot{\mathbf{q}}, \quad \dot{\mathbf{p}} = \mathbf{B}\dot{\mathbf{q}} \quad (46)$$

with

$$\mathbf{A} \equiv [\mathbf{e}_1 \ \mathbf{e}_2 \ \dots \ \mathbf{e}_n], \quad \mathbf{B} \equiv [\mathbf{e}_1 \times \mathbf{r}_1 \ \mathbf{e}_2 \times \mathbf{r}_2 \ \dots \ \mathbf{e}_n \times \mathbf{r}_n] \quad (47)$$

Therefore, the Jacobian mapping $\dot{\mathbf{q}}$ into $\dot{\mathbf{t}}$ is

$$\mathbf{J} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} = [\mathbf{j}_1 \ \mathbf{j}_2 \ \dots \ \mathbf{j}_n], \quad \mathbf{j}_i = \begin{bmatrix} \mathbf{e}_i \\ \mathbf{e}_i \times \mathbf{r}_i \end{bmatrix} \quad (48)$$

where (3×6) submatrices \mathbf{A} and \mathbf{B} are, respectively, known as the orientation and position Jacobians.

Earlier in this section, we assumed only revolute joints to compute the Jacobian of a serial manipulator. If a i th joint is prismatic instead, the angular and linear velocities of the i th link are written as

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1}, \quad \dot{\mathbf{a}}_i = \boldsymbol{\omega}_{i-1} \times \mathbf{a}_i + \dot{d}_i \mathbf{e}_i \quad (49)$$

We can then prove that the contributing member of the i th joint to the Jacobian, i.e., the i th column, is expressed as

$$\mathbf{j}_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix} \quad (50)$$

Example: Jacobian of a 6-DoF Wrist-Partitioned Serial Manipulator

Since the axes of the last joints of a wrist-partitioned serial manipulator intersect at one point, known as the spherical wrist, its Jacobian matrix is simplified, resulting in

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{0} \end{bmatrix} \quad (51)$$

where $\mathbf{0}$, \mathbf{J}_{11} , \mathbf{J}_{12} and \mathbf{J}_{21} are (3×3) matrices. You should note that to simplify the equations, the Jacobian matrix given here maps the joint rates into the twist of P_w , namely the location of the intersection of the axes of the last three joints. Therefore, we have

$$\mathbf{t}_w = \mathbf{J}\dot{\mathbf{q}} \quad (52)$$

where $\mathbf{t}_w = [\boldsymbol{\omega}^T \ \dot{\mathbf{p}}_w^T]^T$. As you can see, the angular velocity vector $\boldsymbol{\omega}$ is not a function of the location of P_w . The linear velocity of P_w , which is only a function of the first three joint velocities, is computed with the following equation, i.e.,

$$\dot{\mathbf{p}}_w = \dot{q}_1 \mathbf{e}_1 \times \mathbf{r}_1 + \dot{q}_2 \mathbf{e}_2 \times \mathbf{r}_2 + \dot{q}_3 \mathbf{e}_3 \times \mathbf{r}_3 \quad (53)$$

where \mathbf{r}_i is defined as the vector from the i th DH frame to the P_w and \mathbf{e}_i is the unit vector parallel to the axis of the i th joint, as mentioned above. The angular velocity of the EE is computed with the formula given earlier in this section, i.e.,

$$\boldsymbol{\omega} = \dot{q}_1 \mathbf{e}_1 + \dot{q}_2 \mathbf{e}_2 + \dot{q}_3 \mathbf{e}_3 + \dot{q}_4 \mathbf{e}_4 + \dot{q}_5 \mathbf{e}_5 + \dot{q}_6 \mathbf{e}_6 \quad (54)$$

Therefore, we can determine that the submatrices included in expression (51) are

$$\mathbf{J}_{11} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3] \quad (55a)$$

$$\mathbf{J}_{12} = [\mathbf{e}_4 \ \mathbf{e}_5 \ \mathbf{e}_6] \quad (55b)$$

$$\mathbf{J}_{21} = [\mathbf{e}_1 \times \mathbf{r}_1 \ \mathbf{e}_2 \times \mathbf{r}_2 \ \mathbf{e}_3 \times \mathbf{r}_3] \quad (55c)$$

Singularities

In robotics, when a manipulator is in a singular posture, or simply in a singularity, it cannot displace its EE along at least one direction. Mathematically, this corresponds to a **singular Jacobian matrix** use to compute joint velocities. We assumed previously this matrix was non-singular, i.e., for a robot with six DoFs, its Jacobian is invertible and its determinant is not equal to zero (Section 6.4). It might not be the case for certain configurations. Beyond the numerical issue of inverting a singular matrix, the corresponding posture of the robot also has a physical meaning related to the **limits of the workspace** of the robot or a **loss of mobility**, as mentioned above. Moreover, if we refer back to the configuration types discussed earlier in this chapter, the singularities correspond to **boundaries** between these entities within the workspace of the robot.

A posture close to a singularity is also problematic for a manipulator and a robot in general, as the determinant of its Jacobian matrix will be close to zero, yielding a division by a number close to zero. This will result in significantly high joint velocities, which raises safety concerns and reduces the trajectory-tracking accuracy. Let

$$\mathbf{t} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (56)$$

where $\dot{\mathbf{q}}$, \mathbf{t} and $\mathbf{J}(\mathbf{q})$ are, respectively, the n -dimensional joint-rate vector, the six-dimensional EE twist and the $6 \times n$ Jacobian matrix, where n is the number of joints. It is thus trivial to see that any given feasible EE twist, namely its linear and angular velocity, as defined in Sect. 10.4.4, is a linear combination of the joint velocities. To be able to achieve any arbitrary value of \mathbf{t} , the rank of \mathbf{J} , which is a function of the posture of the robot, i.e. \mathbf{q} , must be equal to six for a robot in 3D space. If it is the case, any given twist of the EE is feasible. However, it should be noted that since the Jacobian is posture-dependent, it is not always the case. If the rank(\mathbf{J}) becomes

lower than six, this is called a singular posture, or, for brevity, a singularity. Depending on which part of the Jacobian matrix generates a singularity, we can have a position or an orientation singularity, each having a different physical interpretation.

Singularity of the Position Jacobian

For a 6-DoF wrist-partitioned serial manipulator, a singularity of the submatrix \mathbf{J}_{21} causes a position singularity, corresponding to the impossibility of computing the joint rates for this location. This occurs when the determinant of \mathbf{J}_{21} is equal to zero. Considering Eq. (51), the determinant can be written as

$$\det(\mathbf{J}_{21}) = (\mathbf{e}_1 \times \mathbf{r}_1) \times (\mathbf{e}_2 \times \mathbf{r}_2) \times (\mathbf{e}_3 \times \mathbf{r}_3) = 0 \quad (57)$$

This situation occurs in two situations. First, you will find this type of singularity when one column of \mathbf{J}_{21} is equal to zero, for instance, when \mathbf{e}_i and \mathbf{r}_i are parallel, which is commonly called a *shoulder singularity*. This particular case corresponds physically to the wrist center being located on the first joint axis, resulting in the instantaneous loss of one DoF. It can also be true for the second or third joint (wrist center being located on the i th joint axis), but this is usually avoided by carefully designing the manipulator.

Otherwise, we can also have $\det(\mathbf{J}_{21}) = 0$ when two columns of \mathbf{J}_{21} become coplanar, resulting in a rank-deficiency. Multiple postures/configurations of the robot can lead to this, notably, but not only, a fully extended arm at the limit of the reachable workspace. This includes *elbow singularities*, which occurs for vertically articulated⁸ manipulators such as the Meca500 sold by Mecademics.⁹ when the wrist center lies on the plane passing through the second and third axes. This can also happen in theory with the manipulator folded on itself, but mechanical limits normally prevent this situation from occurring.

Singularity of the Orientation Jacobian

In the case of a wrist-partitioned manipulator, an orientation singularity occurs when $\det(\mathbf{J}_{12}) = 0$. This can only happen when \mathbf{e}_4 , \mathbf{e}_5 and \mathbf{e}_6 are **coplanar**. In this configuration, only angular velocity vector on the plane generated by the three vectors mentioned above are possible at the EE. Considering the typical kinematic chain of a serial wrist-partitioned manipulator, it generally occurs when the axes of the fourth and sixth revolute joints are coincident. This type of singularity is sometimes called a **wrist singularity**.

Singularities with a Non-Wrist-Partitioned Manipulators

We now have seen the different singularities within the workspace of a serial wrist-partitioned manipulator through an analysis of its Jacobian. Mathematically, you

⁸ A vertically articulated architecture is common for commercially available wrist-partitioned six-axis serial manipulators: the axes of the second and third joints are parallel, the axes of the first and fourth joints are orthogonal to the axes of the second and third joints and the axis of fifth joint is orthogonal to the axes of fourth and sixth joints.

⁹ <https://www.mecademic.com/en/what-are-singularities-in-a-six-axis-robot-arm>.

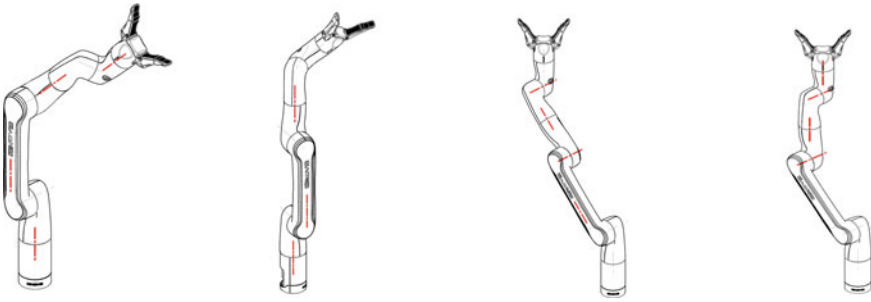


Fig. 10.5 Singular postures of the Kinova Gen3 lite (extracted from user manual)

can apply the same process to find singularities in the workspace of a non-wrist-partitioned manipulator. However, we will look at the full Jacobian matrix in this case, since we do not have decoupled kinematics for the orientation and position. To this aim, we will use the Kinova Gen3 lite previously mentioned to illustrate the process. Potential singular postures are shown in Fig. 10.5.

In this figure, from left to right, we have four different configurations corresponding to singularities of Jacobian matrix that differ from a fully extended arm, another singular configuration. We have, from left to right (all axes mentioned are illustrated in red in the figure),

1. The axis of the first joint and the X -axis of the third DH frame, i.e., the common perpendicular between the axes of joints 2 and 3, are parallel; the axes of joints 4 and 6 are also parallel.
2. The axes of the first and fourth joints are both parallel to the common perpendicular between the axes of joints 2 and 3.
3. The axes of the third and fifth joints are parallel; the fourth joint is also parallel to the common perpendicular between the axes of joints 2 and 3.
4. The axis of the third joint is parallel with the fifth joint axis and the fourth joint axis is parallel with the sixth joint axis.

All four cases illustrated above involve a double alignment in the posture of the Gen3 Lite, which loses a DoF momentarily. For example, in the second case, the EE cannot move in the direction of the fourth joint axis. In the third and fourth cases, motion is impossible in the direction of the axis of the third joint.

Kinematics of Parallel Manipulators

As we mentioned at the beginning of this chapter, parallel manipulators are known for their structural rigidity, speed and the ability to lift a larger payload compared to serial manipulators with similar mass and size. While their architecture is composed of at least one loop, they commonly have more. Among the well-known parallel architectures, the three-limb *Delta* (sometimes with a telescopic Cardan shaft to add a fourth DoF) (Clavel, 1990) as well as the four-limb *Par4* (Pierrot et al., 2003)

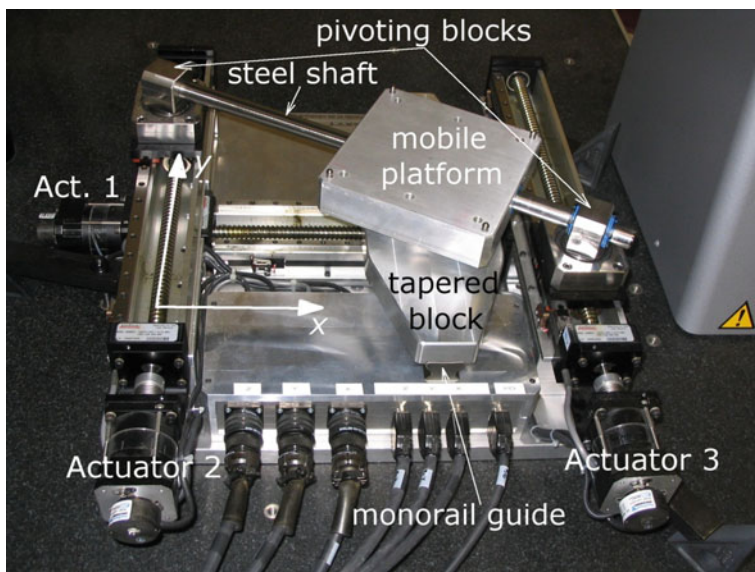


Fig. 10.6 PPR-2PRP parallel robot, from (Joubair et al., 2012)

(*Adept Quattro*) have been patented and commercialized. Before starting with the kinematics of parallel manipulators, you should know that the EE of a parallel robot is commonly called the mobile (or moving) platform, considering it is attached to the base with several limbs.

Direct and Inverse Kinematics

While solving the forward kinematics of a serial kinematic chain is generally a simple task, it is not the case with parallel robots. Indeed, the tool we used in Sect. 10.4.2, the Denavit-Hartenberg convention, is not appropriate for parallel manipulators, as it only accepts a maximum of two joints for each link. In general, it is not possible to obtain an explicit function of the Cartesian coordinates of the EE with respect to the joint coordinates, even for a simple parallel robot. Therefore, iterative methods are commonly used for this purpose.

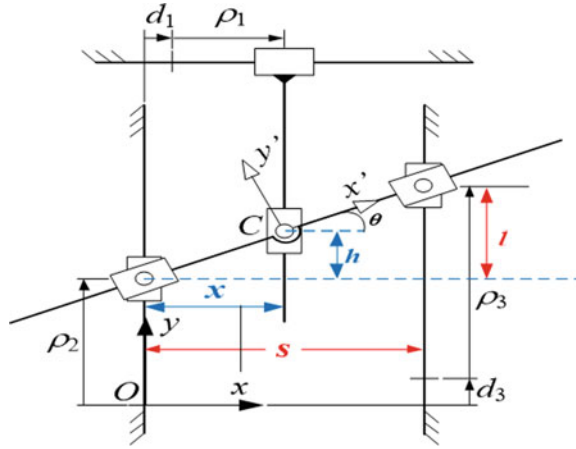
Contrary to the forward kinematics, solving the IKP of a parallel robot is usually less challenging than with a serial robot. We will obtain an implicit function equal to zero where \mathbf{q} and \mathbf{p} are the variables, i.e.,

$$\mathbf{f}(\mathbf{q}, \mathbf{p}) = \mathbf{0} \quad (58)$$

Example: Kinematics of a PPR-2PRP Parallel Robot

Here is a planar parallel robot with three prismatic actuated joints connected to three limbs attached to the mobile platform, shown in Figs. 10.6 and 10.7. One is a PPR

Fig. 10.7 Geometry of a PPR-2PRP parallel robot, from (Joubair et al., 2012)



chain, while the other two are PRP chains. The mobile platform's coordinates are (x, y, θ) , and the joint coordinates are (ρ_1, ρ_2, ρ_3) . We thus need to find expressions of the former as a function of the latter. Using simple geometric relationships, we have:

$$\theta = \arctan\left(\frac{\rho_3 + d_3 - \rho_2}{s}\right) \quad (59a)$$

$$x = \rho_1 + d_1 \quad (59b)$$

For the last Cartesian coordinate, knowing

$$\frac{h}{x} = \frac{l}{s} \quad (60)$$

we can compute

$$y = \rho_2 + (\rho_1 + d_1) \frac{\rho_3 + d_3 - \rho_2}{s} \quad (61)$$

These three expressions above represent the solution to the FKP. The solution to the IKP is straightforward from this point:

$$\rho_1 = x - d_1 \quad (62a)$$

$$\rho_2 = y - x \tan \theta \quad (62b)$$

$$\rho_3 = y + (s - x) \tan \theta - d_3 \quad (62c)$$

Jacobians

As mentioned above, the kinematics model of a parallel robot is generally expressed as an implicit function, namely Eq. (58). By differentiating it with respect to time, we have

$$\mathbf{J}\dot{\mathbf{p}} = \mathbf{K}\dot{\mathbf{q}} \quad (63)$$

where both \mathbf{J} and \mathbf{K} are Jacobian matrices.

Singularities

From these two Jacobian matrices, we can define three types of singularities:

1. Type I: When \mathbf{K} is singular, i.e., $\det(\mathbf{K}) = 0$. This usually corresponds to a limit of the reachable workspace or an internal limit of the workspace where two branches of solutions to the IKP meet. Therefore, certain Cartesian velocities at the EE will not be possible to generate.
2. Type II: When \mathbf{J} is singular, i.e., $\det(\mathbf{J}) = 0$. These singularities occur at locations within the reachable workspace where two branches of solutions to the FKP meet. Therefore, even for a fixed joint coordinates, an infinitesimal motion of the end-effector is possible. This also means that the robot cannot balance certain external wrenches applied to the EE, thus resulting in a loss of control, which must be absolutely avoided.
3. Type III: A combination of both types above, thus when $\det(\mathbf{J}) = \det(\mathbf{K}) = 0$. In this case, Eq. (58) degenerates, resulting in an unusable EE. This kind of singularity only exists for certain architectures.

Figure 10.8 depicts singular postures of a pantograph, a common five-bar mechanism that can be used as a planar parallel manipulator. The EE is on the middle revolute joint and the two revolute joints attached to the base are actuated. As can be seen in this figure, the EE cannot move further up since the mechanism is fully extended for the illustrated type-I singularity. In the case of the type II singular posture depicted, it is impossible to control the vertical motion of the EE. With a small perturbation, the EE could move up or down for the same velocities of the actuated base joints.

Dynamics

According to the Merriam-Webster dictionary, dynamics is “a branch of mechanics that deals with forces and their relation primarily to the motion but sometimes also to the equilibrium of bodies.”¹⁰ Forces can be linear, but also rotational, namely torque.

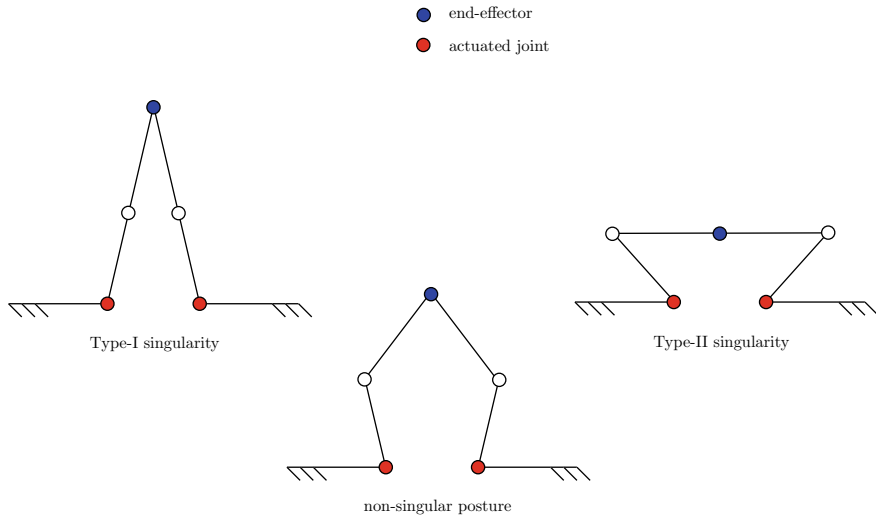


Fig. 10.8 Pantograph, a 2-DoF planar parallel manipulator

The second Newton's law is particularly significant when it comes to the quantitative analysis of the dynamics of a system, as it states that “the time rate of change of the momentum of a body is equal in both magnitude and direction to the force imposed on it.”¹¹ Similarly to kinematics, we can define two different problem:

- forward dynamics, from the actuators to the motion, useful for simulations;
- inverse dynamics, from the motion to the actuators, essential for control.

In this chapter, a brief overview of two approaches to compute the dynamics model of a robot is given, namely the Euler-Lagrange and the Newton-Euler methods.

Euler-Lagrange

The Euler-Lagrange method is based on energy. The *Lagrangian* is defined as

$$\mathcal{L} = \mathcal{T} - \mathcal{V} \quad (64)$$

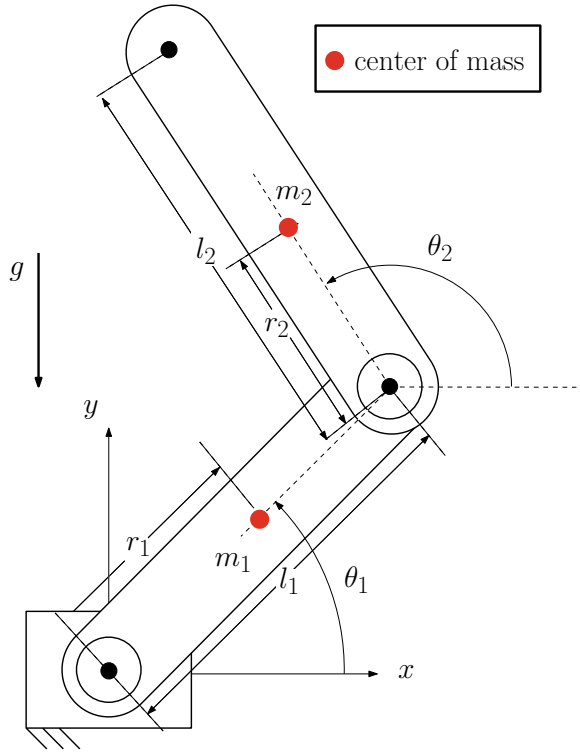
where T and V are, respectively, the total kinetic and potential energies in the system. From the Lagrangian, the dynamics equations defining the robot's motion are computed with

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \quad (65)$$

¹⁰ www.merriam-webster.com/dictionary/dynamics.

¹¹ Definition from www.britannica.com/science/Newtons-laws-of-motion.

Fig. 10.9 Geometry of a 2-DoF serial robot



where the individual q_i and τ_i are, respectively, the generalized joint coordinates and torque (or force for a prismatic joint).

Example: Euler-Lagrange Applied to a 2-DoF Planar Manipulator

A simple 2-DoF serial planar manipulator is illustrated in Fig. 10.9. For the purpose of this example, only the mass of each link is considered and not their moment of inertia. The expression of total kinetic is

$$K = K_1 + K_2 = \frac{1}{2}m_1v_1^2 + \frac{1}{2}m_2v_2^2 \quad (66)$$

where v_1 and v_2 are the magnitude of the linear velocity of masses m_1 and m_2 , respectively. We know, considering the geometry, that

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 = r_1^2 \dot{\theta}_1^2 \quad (67a)$$

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 \quad (67b)$$

$$\dot{x}_2 = (-l_1 \sin \theta_1 - r_2 \sin(\theta_1 + \theta_2))\dot{\theta}_1 - r_2 \sin(\theta_1 + \theta_2)\dot{\theta}_2 \quad (67c)$$

$$\dot{y}_2 = (l_1 \cos \theta_1 + r_2 \cos(\theta_1 + \theta_2))\dot{\theta}_1 + r_2 \cos(\theta_1 + \theta_2)\dot{\theta}_2 \quad (67d)$$

where m_1 , r_1 , l_1 , l_2 , r_2 , m_2 and g are, respectively, the masses, distances between the origin of each link and its CoM and lengths of the first and second links and the gravitational acceleration. The total kinetic energy is thus

$$K = \frac{1}{2}m_1r_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2((l_1^2 + 2l_1l_2\cos\theta_2 + l_2^2)\dot{\theta}_1^2 + 2(l_2^2 + l_1l_2\cos\theta_2)\dot{\theta}_1\dot{\theta}_2 + l_2^2\dot{\theta}_2^2) \quad (68)$$

Finally, again considering the geometry, the total potential energy is

$$T = T_1 + T_2 = m_1gl_1\sin\theta_1 + m_2g(l_1\sin\theta_1 + l_2\sin(\theta_1 + \theta_2)) \quad (69)$$

You can complete the procedure as an exercise.

Newton-Euler

The Newton-Euler approach is a recursive method. You first compute the angular and linear velocities and accelerations of each link individually in the inertial frame, starting from the base. Then, the forces and torques applied by each link on the previous one are computed, starting from the end-effector. It is used here to solve the inverse dynamics of serial manipulators.

Velocities and Accelerations

First, it should be noted that in this procedure, it is the velocity and acceleration of the center of mass (CoM) of each body, not frame, that you need to compute. The velocities and accelerations are obtained with the Algorithm 1. In this table, the components of vectors \mathbf{a}_i and \mathbf{e}_i (cf. Fig. 10.2) in frame $(i + 1)$ are

$$[\mathbf{e}_i]_{i+1} = [0 \sin\alpha_i \cos\alpha_i]^T \quad (70a)$$

$$[\mathbf{a}_i]_{i+1} = [a_i \ b_i \sin\alpha_i \ b_i \cos\alpha_i]^T \quad (70b)$$

Forces and Moments

The next step is to compute the forces and moments on each link, starting with the EE. The wrench applied by the $(i - 1)$ th link on the i th link is defined as

$$\mathbf{w}_i = [\mathbf{n}_i^T \ \mathbf{f}_i^T]^T \quad (71)$$

where the three-dimensional vectors \mathbf{n}_i^T and \mathbf{f}_i^T , are, respectively, the force and moment associated to this wrench. One component of each wrench is the actuation associated to the corresponding joint, namely the third component for a revolute joint and the sixth joint for a prismatic joint. The remaining components are the reaction force and moment between the two links. The procedure to compute the

Algorithm 1 Velocities and accelerations**Require:** $[\omega_0]_1, [\dot{c}_0]_1, [\ddot{c}_0]_1$ and $[\ddot{c}_0]_1$ **for** $i = 1$ to n **do** **if** i th joint is revolute **then**

$$[\omega_i]_{i+1} \leftarrow Q_i^T [\omega_{i-1}]_i + \dot{\theta}_i [\mathbf{e}_i]_{i+1}$$

$$[\dot{c}_i]_{i+1} \leftarrow Q_i^T [\dot{c}_{i-1}]_i + [\omega_i]_{i+1} \times [(\mathbf{a}_i + \mathbf{s}_i)]_{i+1} - Q_i^T [\omega_{i-1} \times \mathbf{s}_{i-1}]_i$$

$$[\dot{\omega}_i]_{i+1} \leftarrow Q_i^T [\dot{\omega}_{i-1}]_i + \ddot{\theta}_i [\mathbf{e}_i]_{i+1} + \dot{\theta}_i (Q_i^T [\omega_{i-1}]_i) \times [\mathbf{e}_i]_{i+1}$$

$$[\ddot{c}_i]_{i+1} \leftarrow Q_i^T [\ddot{c}_{i-1}]_i + [\dot{\omega}_i]_{i+1} \times [(\mathbf{a}_i + \mathbf{s}_i)]_{i+1} + [\omega_i]_{i+1} \times [\omega_i]_{i+1} \times [(\mathbf{a}_i + \mathbf{s}_i)]_{i+1} - Q_i^T [\dot{\omega}_{i-1} \times (\omega_{i-1} \times \mathbf{s}_{i-1})]_i$$

else if i th joint is prismatic **then**

$$[\omega_i]_{i+1} \leftarrow Q_i^T [\omega_{i-1}]_i$$

$$[\dot{c}_i]_{i+1} \leftarrow Q_i^T [\dot{c}_{i-1}]_i + [\omega_i]_{i+1} \times [(\mathbf{a}_i + \mathbf{s}_i)]_{i+1} + \dot{d}_i [\mathbf{e}_i]_{i+1} - Q_i^T [\omega_{i-1} \times \mathbf{s}_{i-1}]_i$$

$$[\dot{\omega}_i]_{i+1} \leftarrow Q_i^T [\dot{\omega}_{i-1}]_i$$

$$[\ddot{c}_i]_{i+1} \leftarrow Q_i^T [\ddot{c}_{i-1}]_i + [\dot{\omega}_i]_{i+1} \times [(\mathbf{a}_i + \mathbf{s}_i)]_{i+1} + [\omega_i]_{i+1} \times [\omega_i]_{i+1} \times [(\mathbf{a}_i + \mathbf{s}_i)]_{i+1} - Q_i^T [\dot{\omega}_{i-1} \times (\omega_{i-1} \times \mathbf{s}_{i-1})]_i + 2[\omega_i]_{i+1} \times \dot{d}_i [\mathbf{e}_i]_{i+1} + \ddot{d}_i [\mathbf{e}_i]_{i+1}$$

end if**end for**

wrench on each link is detailed in Algorithm 2. You may wonder where the effect of gravity appears in the algorithm. To simplify the procedure while still obtaining an equivalent solution, we use a simple trick. Here, we suppose a virtual acceleration $-\mathbf{g}$ at the base of the robot, namely the first link. Therefore, even though the base is fixed and not moving, we have

$$[\ddot{c}_0]_1 = [-\mathbf{g}]_1 \quad (72)$$

where $-\mathbf{g}$ is the gravitational acceleration.

Algorithm 2 Wrench on each link

$$[\mathbf{f}_n]_n \leftarrow Q_n [m_n \ddot{c}_n - \mathbf{f}]_{n+1}$$

$$[\mathbf{n}_n]_n \leftarrow Q_n [\mathbf{I}_n \dot{\omega}_n + \omega_n \times \mathbf{I}_n \omega_n - \mathbf{n} + (\mathbf{a}_n + \mathbf{s}_n) \times \mathbf{f}_n]_{n+1}$$

for $i = n - 1$ to 1 **do**

$$[\mathbf{f}_i]_i \leftarrow Q_i [m_i \ddot{c}_i + \mathbf{f}_{i+1}]_{i+1}$$

$$[\mathbf{f}_i]_{i+1} \leftarrow Q_i [\mathbf{f}_i]_i$$

$$[\mathbf{n}_i]_i \leftarrow Q_i [\mathbf{I}_i \dot{\omega}_i + \omega_i \times \mathbf{I}_i \omega_i + \mathbf{n}_{i+1} + (\mathbf{a}_i + \mathbf{s}_i) \times \mathbf{f}_i - \mathbf{s}_i \times \mathbf{f}_{i+1}]_{i+1}$$

end for

We now have all forces \mathbf{f}_i and moments \mathbf{n}_i ; the final step is thus to compute what we were looking for at the beginning, the actuation torques for revolute joints and actuation forces for prismatic joint. This is done with the following two equations:

$$\tau_i = \mathbf{e}_i^T \mathbf{n}_i, \quad \text{for revolute joints} \quad (73a)$$

$$\tau_i = \mathbf{e}_i^T \mathbf{f}_i, \quad \text{for prismatic joints} \quad (73b)$$

Chapter Summary

In this chapter, an introduction to the fundamentals of robotics manipulators, from the mechanics point-of-view, was given. We first introduced the typical architectures, serial and parallel, their pros and cons, as well as notable characteristics such as their DoFs and the type of motion they can generate. Then, we focused on the kinematics of both categories, from the joints to the end-effector (direct kinematics) and the other way around (inverse kinematics). While standard approaches exist for both the forward and inverse kinematics of a serial manipulator, notably if it is wrist-partitioned, it is not the case for their parallel counterparts. However, the IKP of a parallel robot can generally be solved more easily. We have studied the relations between the joint velocities and the twist of the EE, which includes its angular and linear velocities. These equations can be put together to obtain the Jacobian matrix, an useful tool in the analysis of serial and parallel robots. Indeed, from this matrix, singular postures of the robot can be found: these configurations must be avoided, because they may cause safety and control issues. Finally, we did a brief overview of the dynamics of robotic manipulators, namely two common approaches, Euler-Lagrange and Newton-Euler.

Revision Questions

Question #1

Which equations are valid (there could be more than one or none)?

1. $\mathbf{H}_{tool}^{workshop} = \mathbf{H}_{workshop}^0 \mathbf{H}_1^1 \mathbf{H}_2^2 \mathbf{H}_3^3 \mathbf{H}_{tool}^3$
2. $\mathbf{H}_{tool}^{workshop} = \mathbf{H}_{workshop}^0 \mathbf{H}_1^0 \mathbf{H}_2^1 \mathbf{H}_3^2 \mathbf{H}_0^{workshop}$
3. $\mathbf{H}_3^0 = \mathbf{H}_1^0 \mathbf{H}_2^1 \mathbf{H}_3^2$
4. $\mathbf{H}_0^3 = \mathbf{H}_1^0 \mathbf{H}_2^1 \mathbf{H}_3^2$
5. $\mathbf{H}_{EE}^{workshop} = \mathbf{H}_{workshop}^0 \mathbf{H}_0^1 \mathbf{H}_1^2 \mathbf{H}_2^3 \mathbf{H}_3^{EE}$

Question #2

Inverse kinematics makes it possible to obtain. . .

Please choose an answer:

1. the pose of the robot effector, based on its parameters and joint coordinates;
2. the values of the joint coordinates of the robot, from the pose of the effector and the parameters of the robot;
3. the position of the robot effector, based on its parameters and joint coordinates.

Question #3

The kinematic chains of parallel robots are made of:

1. passive and active joints;
2. only passive joints;
3. only active joints.

Table 10.2 DH parameters of a wrist-partitioned 6R manipulator

i	1	2	3	4	5	6
a_i	0	135 mm	38 mm	0	0	0
d_i	135 mm	0	0	120 mm	0	70 mm
α_i	$-\pi/2$	0	$-\pi/2$	$\pi/2$	$\pi/2$	0
θ_i	q_1	$q_2 - \pi/2$	q_3	q_4	$q_5 + \pi$	q_6

Question #4

Regarding the computation of the Jacobian matrix of a serial manipulator, the vector $\mathbf{e}_{i-1}^{workshop}$ represents:

1. the unit vector parallel to the X -axis of the $(i - 1)$ th frame with respect to the workshop;
2. the unit vector parallel to the Y -axis of the $(i - 1)$ th frame with respect to the workshop;
3. the unit vector parallel to the Z -axis of the $(i - 1)$ th frame with respect to the workshop.

Question #5

The DH parameters of a wrist-partitioned manipulator are given in Table 10.2. First, compute the six homogeneous transformations matrices. Then, compute the solution(s) to the inverse kinematics for a Cartesian position of the wrist of (250, 0, 150) mm and an orientation with the Euler angles of $(0^\circ, 90^\circ, 0^\circ)$ according to the XYZ mobile convention.

Further Reading

This chapter only gave you a short summary on the mechanics of robotic manipulators. If you want to learn more, you can first take a look into the original DH notation and its variants. To this aim, you can refer to a paper published by Harvey Lipkin (Lipkin, 2008). Moreover, given the fact that we did not go into the details of the dynamics of robots, extensive literature can be found on this topic. Notably, you can look into the Kane's equations, similar to Lagrangian approach. Also, (Angeles, 2014) introduced an alternative method to solve the inverse dynamics of a robotic system, the *natural orthogonal complement* (NOC). Regarding mathematical tools useful for the analysis of the mobility, kinematics and dynamics of robotic systems and mechanisms, you can take a look into group theory (Angeles, 2014), screw theory (Davidson, 2004; Müller, 2017), where the twist and wrench concept originate, and dual-numbers algebra, useful to combine a translation and a rotation into one

single variable. Finally, you can also look into the concept of *constraint singularities* for parallel mechanisms (Zlatanov et al., 2002).

References

- Angeles, J. (2014). *Fundamentals of Robotic Mechanical Systems, Mechanical Engineering Series* (Vol. 124). Springer International Publishing. <https://doi.org/10.1007/978-3-319-01851-5>
- Clavel, R. (1990). *Device for the movement and positioning of an element in space*
- Davidson, J. K. (2004). *Robots and screw theory?: Applications of kinematics and statics to robotics*. Oxford University Press.
- Forsythe, G. E. (1970). Pitfalls in Computation, or Why a Math Book isn't Enough. *The American Mathematical Monthly*, 77(9), 931–956. <https://doi.org/10.1080/00029890.1970.11992636>
- Gosselin, C., & Liu, H. (2014). Polynomial Inverse Kinematic Solution of the Jaco Robot. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, Buffalo, NY, p. V05BT08A055. <https://doi.org/10.1115/detc2014-34152>
- Hartenberg, R., & Denavit, J. (1964). *Kinematic synthesis of linkages*. McGraw-Hill Book Company.
- Joubair, A., Slamani, M., & Bonev, I. A. (2012). A novel XY-Theta precision table and a geometric procedure for its kinematic calibration. *Robotics and Computer-Integrated Manufacturing*, 28(1), 57–65. <https://doi.org/10.1016/J.RCIM.2011.06.006>
- Lipkin, H. (2008). A note on Denavit-Hartenberg notation in robotics. In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, DETC2005, Vol. 7B, pp. 921–926. <https://doi.org/10.1115/DETC2005-85460>
- Makino, H., Kato, A., & Yamazaki, Y. (2007). Research and commercialization of SCARA Robot: The case of industry-university joint research and development. *International Journal of Automation Technology*, 1, 61–67.
- Müller, A. (2017). Screw theory: A forgotten Tool in Multibody Dynamics. *PAMM*, 17(1), 809–810. <https://doi.org/10.1002/PAMM.201710372>
- Pierrot, S., Morita Pierrot, F., Shibukawa, T., & Morita, K. (2003). *Four-degree-of-freedom parallel robot*
- Zlatanov, D., Bonev, I. A., & Gosselin, C. M. (2002). Constraint singularities of parallel mechanisms. *Proceedings—IEEE International Conference on Robotics and Automation*, 1, 496–502. <https://doi.org/10.1109/ROBOT.2002.1013408>

Bruno Belzile is a postdoctoral fellow at the INIT Robots Lab. of ÉTS Montréal in Canada. He holds a B.Eng. degree and Ph.D. in mechanical engineering from Polytechnique Montréal. His thesis focused on underactuated robotic grippers and proprioceptive tactile sensing. He then worked at the Center for Intelligent Machines at McGill University, where his main areas of research were kinematics, dynamics and control of parallel robots. At ÉTS Montréal, he aims at creating spherical mobile robots for planetary exploration, from the conceptual design to the prototype.

David St-Onge (Ph.D., Mech. Eng.) is an Associate Professor in the Mechanical Engineering Department at the École de technologie supérieure and director of the INIT Robots Lab (initrobots.ca). David's research focuses on human-swarm collaboration more specifically with respect to operators' cognitive load and motion-based interactions. He has over 10 years' experience in the field of interactive media (structure, automatization and sensing) as workshop production director and as R&D engineer. He is an active member of national clusters centered on human-robot interaction (REPARTI) and art-science collaborations (Hexagram). He participates in national training programs for highly qualified personnel for drone services (UTILI), as well as for the deployment of industrial cobots (CoRoM). He led the team effort to present the first large-scale symbiotic integration of robotic art at the IEEE International Conference on Robotics and Automation (ICRA 2019).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

