

New Perspectives on Robotics with Geometric Calculus



Isiah Zaplana

Abstract One of the most successful applications of geometric calculus to engineering refers to robotics and computer vision. In this line, this chapter presents an overview of the main classical problems in robot kinematics and motion planning and explains how geometric calculus has been used to solve them by exploiting their algebraic and geometric properties (such as, for instance, that every isometry can be compactly represented, the geometric covariance, the properties of the rotor group and the bivector algebra). Besides, it also introduces recent open problems in robotics and explains how geometric calculus can be used to contribute to their solutions.

1 Introduction

Nowadays, robotics is a well-known field. From science-fiction to real industry, during the last years we have thought and talked a lot about robots and their role in our modern society. But, what exactly is a robot? It is a programmable machine able to carry out a complex series of tasks with some level of autonomy [17]. This definition, however, is too broad for the purposes of this chapter since it includes robots like the vacuum Roomba robot and the Lego toy robot. Here, instead, we are going to focus on *serial industrial robots*, which are of fundamental importance in the industry. They perform tasks that human operators cannot, such as carrying heavy objects, painting, grasping, moving and handling large pieces, etc. Either by assisting human operators or by completely replacing them, serial industrial robots have turned out to be an indispensable element of modern industry.

Geometrically speaking, these robots are sequences of rigid-bodies (called *links*) connected by means of motor-actuated kinematic pairs (called *joints*). Every joint provides relative motion between the two consecutive links it connects (see Fig. 1). The most important point is the free end of the last link, the so-called *end-effector*. Its importance relies on the fact that every tool the robot needs to perform its tasks—painting tools, screw-drives, robotic hands, grippers, etc.—is placed at the

I. Zaplana (✉)

Department of Mechanical Engineering, KU Leuven, Leuven, Belgium
e-mail: isiah.zaplana@kuleuven.be

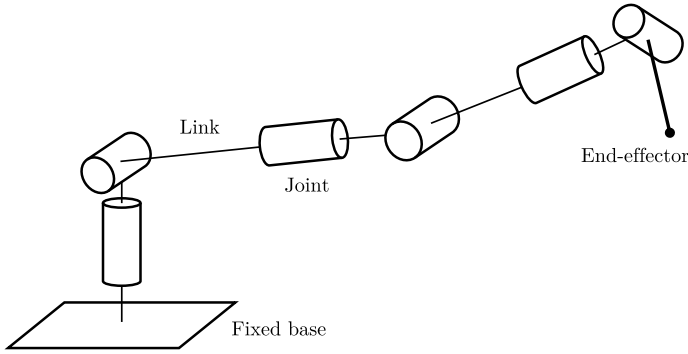


Fig. 1 General scheme of a serial industrial robot

end-effector and, therefore, it is fundamental to know: (1) where the end-effector is at each *configuration* of the entire robot, i.e., its position and orientation in \mathbb{R}^3 at each configuration, and (2) how to move the robot so its end-effector arrives in a predefined desired position and orientation. The first problem is known as the *forward kinematics* problem of a serial industrial robot, while the second is known as the *motion planning* problem. Since both problems analyze the motion of a serial industrial robot without considering the dynamics of the system, they are said to be *kinematic* problems or, more precisely, *robotic kinematic* problems.

This chapter provides a formal mathematical introduction to both problems and develops some tools based on conformal geometric algebra to solve them. In particular, we are going to deal with the forward and inverse kinematics, where the latter is a non-trivial kinematic subproblem of the motion planning problem, as well as with the motion planning problem itself. The rest of the chapter is organized as follows: Sect. 2 formulates the forward and inverse kinematics for general serial industrial robots using conformal geometric algebra and shows how to solve them using this mathematical framework, while the same is done with the motion planning problem in Sect. 3. Finally, we present the conclusions and some open problems in the final section, Sect. 4.

2 Forward and Inverse Kinematics

As stated before, robot kinematics is about studying the motion of general robots without considering the dynamics of the system. In particular, for serial industrial robots, it entails the study of two well-differentiated problems: the forward kinematics and the inverse kinematics. Before we formally formulate both problems, we need to introduce some preliminary concepts.

The joints of serial industrial robots are of two types: *revolute*, if their motion is rotational, and *prismatic*, if their motion is translational. The amount of such motion

is known as the *joint variable* and is denoted by q . Then, for every joint $1 \leq i \leq n$, q_i is either an angle, θ_i , if joint i is revolute or a displacement, d_i , if joint i is prismatic.

Definition 1. The vector of all joint variables $\mathbf{q} = (q_1, \dots, q_n)$ is said to be the *configuration* of the robot. The space of all configurations of a robot is called the *configuration* or *joint space* of the robot and is denoted by C .

A frame $\{\mathbf{o}, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$ is attached to the end-effector of the robot. The three-dimensional point $\mathbf{o} \in \mathbb{R}^3$ describes the position of the end-effector, while the right-handed linear frame $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ describes its orientation. We will use this notation through the rest of the chapter to avoid confusion, so a linear frame will be a set of three mutually orthogonal unitary vectors \mathbf{x} , \mathbf{y} and \mathbf{z} , while a frame will be the pair formed by a three-dimensional point and a linear frame.

Definition 2. The space of all positions and orientations of the end-effector with respect to a reference frame is called the *operational space* of the robot and is denoted by \mathcal{X} . Clearly, $\mathcal{X} \subset SE(3)$, where $SE(3)$ denotes the three-dimensional special Euclidean group.

Definition 3. A serial industrial robot is said to have n *degrees of freedom* (DoF) if its configuration is specified by n joint variables.

Given a configuration $\mathbf{q} \in C$, we want to find the position and orientation of the end-effector associated with that configuration. This is known as the forward kinematics problem. Conversely, the inverse kinematics problem consists of finding the configurations associated with a predefined position and orientation of the end-effector. In other words, the forward kinematics problem consists on finding the continuous function f that assigns the position and orientation \mathbf{x} of the end-effector to each configuration \mathbf{q} :

$$\begin{aligned} f : C &\rightarrow \mathcal{X} \\ \mathbf{q} &\mapsto \mathbf{x} \end{aligned} \tag{1}$$

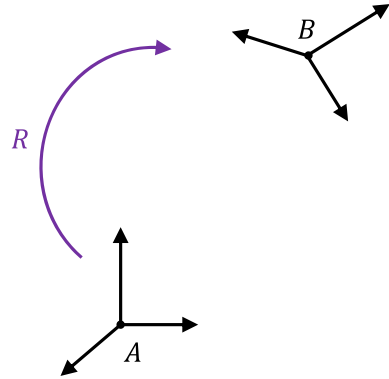
Since f is well-defined, the forward kinematics is said to have *analytical solution*. However, f has not, in general, a global inverse [2, 3] and, as a consequence, the inverse kinematics problem of an arbitrary serial industrial robot has no analytical solution. Hence, geometric and numerical methods need to be developed to solve it.

2.1 Forward Kinematics

In practice, we can also attach a frame to each joint of the robot, that is, for every $1 \leq i \leq n$, we have a frame $\{\mathbf{o}_i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i\}$ attached to joint i . They are known as the *joint frames* of the robot.

Each one of these joint frames depends on the position and orientation of the previous joints, i.e., it depends on the previous joint frames. In particular, for $2 \leq$

Fig. 2 Given two different frames, there is always a rotor relating one to the other. In this case, such a rotor R describes a screw or helical motion (translation followed by a rotation around the same axis) between the two frames



$i \leq n$, the i -th joint frame is related to the $(i - 1)$ -th joint frame, that works as a reference frame. The case $i = 1$ is a special case since there is no previous joint. Hence, the first joint frame has, as a reference frame, the frame placed at the base of the robot, the so-called *base frame*. It is fixed, i.e., it does not depend on the configuration of the robot. Due to this, the base frame is usually considered the global reference frame of the robot.

Each joint frame is determined, not only by the joint variable q_i , but also by the following set of rules [16]:

- The z_i -axis is aligned with the rotational/translational joint axis.
- The x_i -axis is aligned with the common perpendicular to z_i and z_{i-1} , where the latter is the z -axis of the $(i - 1)$ -th joint frame.
- The origin \mathbf{o}_i is set at the intersection of z_i with the common perpendicular to z_i and z_{i-1} .

It is well-known that, given any two linear frames $\{x, y, z\}$ and $\{x', y', z'\}$, there always exists a rotor $R \in \mathcal{G}_3^+$, uniquely determined up to sign, such that:

$$\begin{cases} \mathbf{x}' = R\mathbf{x}\tilde{R} \\ \mathbf{y}' = R\mathbf{y}\tilde{R} \\ \mathbf{z}' = R\mathbf{z}\tilde{R} \end{cases} \quad (2)$$

In conformal geometric algebra, since translations are also encoded by rotors (see section 2.4 of chapter 2 in [13]), we can extend the result to include also their respective origins (see Fig. 2).

Theorem 1. *Given two arbitrary frames $\{\mathbf{o}, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$ and $\{\mathbf{o}', \mathbf{x}', \mathbf{y}', \mathbf{z}'\}$, there always exists a rotor $R \in \mathcal{G}_{4,1}$, uniquely determined up to sign, satisfying that:*

$$\begin{cases} \mathbf{o}' = R\mathbf{o}\tilde{R} \\ \mathbf{x}' = R\mathbf{x}\tilde{R} \\ \mathbf{y}' = R\mathbf{y}\tilde{R} \\ \mathbf{z}' = R\mathbf{z}\tilde{R} \end{cases} \quad (3)$$

In addition, $R = R_1R_2$, where:

$$R_1 = 1 - \frac{\mathbf{v}e_\infty}{2}, \quad (4)$$

with $\mathbf{v} = \mathbf{o}' - \mathbf{o}$ and

$$R_2 \in \mathcal{G}_3^+, \quad (5)$$

i.e., R_1 is the rotor encoding the translation that maps \mathbf{o} to \mathbf{o}' and R_2 is the rotor encoding the rotation that transforms $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ to $\{\mathbf{x}', \mathbf{y}', \mathbf{z}'\}$.

As a corollary of Theorem 1, we have that we can recover the rotor that transforms the $(i - 1)$ -th joint frame into the i -th joint frame for every $1 \leq i \leq n$. These rotors allow to relate the base frame of the robot with the end-effector, thus allowing to compute its position and orientation with respect to the global reference frame—i.e., the base frame. Notice that, since the joint frames are determined by the joint variables, so are the rotors recovered from them.

Now, given two consecutive joint frames, how can we construct the rotor that transforms one into the other? There are different ways. For instance, in [13, chapter 4], we followed the standard convention in robot kinematics (known as the Denavit-Hartenberg convention) to construct each intermediate joint frame by means of a set of four parameters (the Denavit-Hartenberg parameters) and, with those parameters, recover the rotor relating each frame to the following one. Here, we are going to follow a different approach based on the use of the reciprocal frame.

Definition 4. Given a linear frame $\{e_1, e_2, e_3\}$, its *reciprocal frame* $\{e^1, e^2, e^3\}$ is the linear frame satisfying that $e_i \cdot e^j = \delta_{ij}$, where $\delta_{(\cdot)}$ denotes the Kronecker delta.

In [7], the reader can find some useful properties of reciprocal frames as well as an explicit expression for calculating them. In addition, there is an expression for the rotor relating two different linear frames $\{e_1, e_2, e_3\}$ and $\{f_1, f_2, f_3\}$:

$$R = \frac{1 + f_1e^1 + f_2e^2 + f_3e^3}{|1 + f_1e^1 + f_2e^2 + f_3e^3|}. \quad (6)$$

If we apply Eq. (6) to the joint frames, we have that:

$$R_{i-1}^i = \frac{1 + \mathbf{x}_i\mathbf{x}^{i-1} + \mathbf{y}_i\mathbf{y}^{i-1} + \mathbf{z}_i\mathbf{z}^{i-1}}{|1 + \mathbf{x}_i\mathbf{x}^{i-1} + \mathbf{y}_i\mathbf{y}^{i-1} + \mathbf{z}_i\mathbf{z}^{i-1}|} \quad (7)$$

is the rotor that transform the $(i - 1)$ -th linear joint frame into the i -th linear joint frame. Now, taking the rotor defined in Eq. (4) and applying it to the origins of these joint frames, we have that:

$$T_{i-1}^i = 1 + \frac{e_{\infty} \mathbf{v}}{2}, \quad (8)$$

where $\mathbf{v} = \mathbf{o}_i - \mathbf{o}_{i-1}$.

In summary, the rotor that transforms the $(i - 1)$ -th joint frame into the i -th joint frame is:

$$M_{i-1}^i = T_{i-1}^i R_{i-1}^i. \quad (9)$$

Hence, the rotor that relates the base frame with the end-effector for a specific configuration $\mathbf{q} \in C$ is:

$$M = M_0^1 M_1^2 \cdots M_{n-1}^n. \quad (10)$$

Therefore, if we take the base frame, i.e., the global reference frame $\{\mathbf{o}, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$, and we applied M to each one of its elements (with the sandwiching product $M(\cdot)\tilde{M}$), we get the frame attached to the end-effector, i.e., its position and orientation with respect to that global reference frame. Again, all we have done so far is configuration-dependent, which means that M is the rotor that solves the forward kinematics problem for a specific $\mathbf{q} \in C$. If we change the configuration, we will need to construct the new joint frames and recover the associated rotors to compute its product. In general, we have the following result.

Theorem 2. *For every configuration $\mathbf{q} = (q_1, \dots, q_n) \in C$, the position and orientation of the end-effector associated with \mathbf{q} is:*

$$P' = M(\mathbf{q})P\tilde{M}(\mathbf{q}) = M_0^1(q_1) \cdots M_{n-1}^n(q_n)P\tilde{M}_{n-1}^n(q_n) \cdots \tilde{M}_0^1(q_1), \quad (11)$$

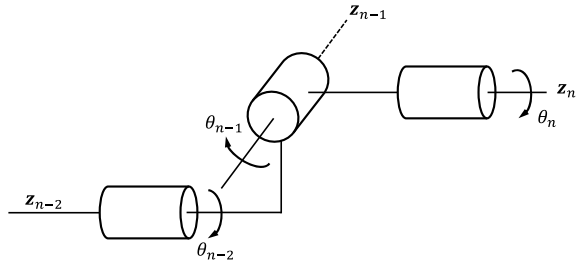
where P is the position (orientation) of the base frame and P' denotes the position (orientation) of the end-effector.

The advantages of the approach presented here include a compact formulation of both the forward kinematics problem and its solution. In addition, since rotors are elements of the algebra, the manipulation of complex geometric structures (like serial chains) becomes easier.

2.2 Inverse Kinematics

A serial industrial robot is said to have a *spherical wrist* if their last three joint axes either intersect at a single point or are parallel. It is well-known that serial industrial robots with a spherical wrist have always analytical or closed-form solutions (by Pieper's theorem [15]). In addition, the proof of Pieper's theorem is constructive in the sense that closed-form solutions are explicitly derived for any type of robot

Fig. 3 Non-spherical wrist. Notice the offset between the $(n - 2)$ -th and the $(n - 1)$ -th joints



with a spherical wrist. However, if there is an offset between any of the last three joint axes (as shown in Fig. 3), then the robot has no longer a spherical wrist and, hence, Pieper's theorem cannot be applied. To solve the inverse kinematics problem for those robots, Paul [14] developed a method based on the homogeneous matrices T_{i-1}^i used to describe the kinematics of serial industrial robots [16]. Indeed, given the kinematic identity:

$$T_0^1 \cdot T_1^2 \cdots T_{n-1}^n = T_0^n, \quad (12)$$

where we recognize in T_0^n the homogeneous matrix describing the position and orientation of the end-effector with respect to the base frame and where T_{i-1}^i only depends on the joint variable q_i , Paul's method consists of analyzing each one of the following matrix equations:

$$T_{i-1}^i \cdots T_{n-1}^n = (T_{i-2}^{i-1})^{-1} \cdots (T_0^1)^{-1} \cdot T_0^n \quad \text{for } i = 2, \dots, n \quad (13)$$

to isolate known trigonometric equations that can be solved analytically for one or more joint variables. However, the large number of different combinations together with the intricacies for solving analytically arbitrary trigonometric equations makes this method not suitable for kinematic chains of complex geometry. Most of the contributions found in the literature [5, 8, 10] focus either on numerical methods or on particular geometric methods. Although the latter can only be applied to the specific robots they have been designed for, they give the complete set of solutions, contrary to what happens with the former, where only one solution is obtained. Nevertheless, geometric methods are difficult to design, especially for robots without a spherical wrist. This is one of the reasons why conformal geometric algebra turns out to be useful to deal with this problem. For instance, the works [4, 6, 9, 18, 19] solve the inverse kinematics of different type of robots by means of conformal geometric algebra. The idea exploited in all of them is to define different geometric entities whose intersections coincide with the origins of the frames attached to the joints. Those points allow us to recover the joint variables and, hence, the configuration or configurations associated with a predefined position and orientation of the end-effector. Finally, in [1] a numerical method also based on conformal geometric algebra is developed to solve the inverse position problem for arbitrary long serial robots.

In our previous work [13, chapter 4], we used conformal geometric algebra to develop a geometric method to cope with the inverse kinematics of serial industrial robots with a spherical wrist based on the same idea of the works cited above. Here, we will go one step further by considering robots without a spherical wrist. As stated before, the most typical case is when there is an offset between any of the last three joint axes. In that case, we cannot split the problem in the two classical subproblems, namely the inverse position problem and the inverse orientation problem (as we did in [13, chapter 4]). This is the first time, to the author's knowledge, that the inverse kinematics of this particular type of robots is addressed by means of conformal geometric algebra.

Let us suppose that the predefined position is denoted by $\mathbf{p} \in \mathbb{R}^3$, while the predefined orientation is denoted by $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$. First, we compute the null vector $p \in \mathcal{G}_{4,1}$ associated with \mathbf{p} :

$$p = H(\mathbf{p}) = \mathbf{p} + e_0 + \frac{1}{2}\mathbf{p}^2 e_\infty, \quad (14)$$

where $H(\cdot)$ denotes the Hestenes' embedding (as defined in section 2.1 of chapter 2 in [13]). With the three vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$ we define three lines, whose inner representation are:

$$\begin{aligned} \ell_x &= \mathbf{x}I - (\mathbf{x} \wedge \mathbf{p})Ie_\infty \\ \ell_y &= \mathbf{y}I - (\mathbf{y} \wedge \mathbf{p})Ie_\infty \\ \ell_z &= \mathbf{z}I - (\mathbf{z} \wedge \mathbf{p})Ie_\infty, \end{aligned} \quad (15)$$

where $I = e_1 \wedge e_2 \wedge e_3$ is the pseudoscalar of \mathcal{G}_3 . Clearly, the orientation of the end-effector is uniquely determined by any two of those lines.

Now, let us consider the serial industrial robot depicted in Fig. 4, i.e., the typical serial industrial robot but, instead of having a spherical wrist, it has an offset between the fourth and the fifth joints. In this case, we have the points p_0 —the null vector representation of the origin of the base frame—and p —the null vector representation of the target position \mathbf{p} . Hence, we need to find the points p_1, p_2 and p_3 .

Point p_1 is the translation of the point p_0 through the z -axis of the first joint frame, \mathbf{z}_1 , a displacement by the length of the first link, a_1 . Therefore:

$$p_1 = T_{\mathbf{z}_1} p_0 \tilde{T}_{\mathbf{z}_1}, \quad (16)$$

where:

$$T_{\mathbf{z}_1} = 1 + \frac{a_1 e_\infty \mathbf{z}_1}{2}. \quad (17)$$

Notice that, since \mathbf{z}_1 is aligned with the first joint axis, it does not change under the action of the joint variable q_1 (in fact, it does not change under the action of any joint variable) and, thus, the base frame and the first joint frame have the same z -axis.

Analogously, point p_3 is the translation of the point p along the z -axis of the last joint, which coincides with the z -axis of the target orientation, a displacement by the length of the last link, a_6 . Hence:

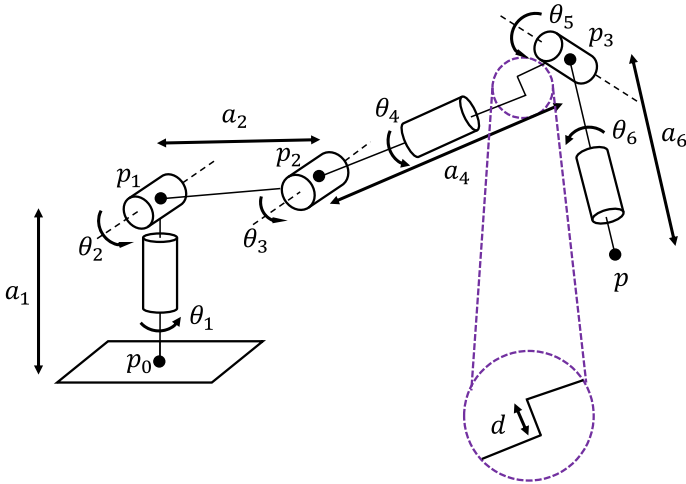


Fig. 4 Schematic representation of a serial industrial robot: the length of the i -th link is denoted by a_i , while the joint variable of the i -th joint is denoted by the angle θ_i . The value d is the length of the offset placed between the fourth and fifth joints (so this robot has not a spherical wrist). To find out the value of the unknown joint variables θ_i , the points p_i , attached to the joints, need to be computed

$$p_3 = T_z p \tilde{T}_z, \quad (18)$$

where:

$$T_z = 1 + \frac{a_6 e_\infty z}{2}, \quad (19)$$

Finally, to compute the point p_2 , we proceed as follows. We need two spheres and one plane, whose inner representations are:

$$\begin{aligned} \pi &= p_0 \wedge p_1 \wedge p_3 \wedge e_\infty \\ s_1 &= p_1 - \frac{1}{2} a_2^2 e_\infty \\ s_2 &= p_3 - \frac{1}{2} (a_4 + d)^2 e_\infty \end{aligned} \quad (20)$$

Clearly, p_2 belongs to the intersection between these three geometric objects.

To compute such a intersection we use the following definition:

Definition 5. Let O_1 and O_2 be two different geometric objects with outer representations K_1^* and K_2^* . The *meet* or *intersection* between O_1 and O_2 , denoted by $K_1^* \vee K_2^*$, is defined as the multivector $K_1^* \vee K_2^* = (K_1 \wedge K_2)^* = K_1 \cdot K_2^*$.

Extended to three geometric objects with outer representations K_1^* , K_2^* and K_3^* , we have that:

$$K_1^* \vee K_2^* \vee K_3^* = (K_1 \wedge K_2 \wedge K_3)^*. \quad (21)$$

Now, since the inner representation of any plane and sphere is a grade-1 element of $\mathcal{G}_{4,1}$, $s_1 \wedge s_2 \wedge \pi$ is a grade-3 element and, as a consequence, its dual is a grade-2 element of $\mathcal{G}_{4,1}$, i.e., a bivector. Therefore:

$$B = s_1^* \vee s_2^* \vee \pi^* = (s_1 \wedge s_2 \wedge \pi)^* \quad (22)$$

is a bivector. This bivector represents a pair of points in conformal geometric algebra, so $B = b_1 \wedge b_2$ for some null vectors b_1 and b_2 . It is clear that p_2 is one of these two null vectors. To extract them from B , the following equations are used [11]:

$$\begin{aligned} b_1 &= -\tilde{P}((b_1 \wedge b_2) \cdot e_\infty) P, \\ b_2 &= P((b_1 \wedge b_2) \cdot e_\infty) \tilde{P}, \end{aligned} \quad (23)$$

where P denotes the projector operator defined as:

$$P = \frac{1}{2} \left(1 + \frac{b_1 \wedge b_2}{|b_1 \wedge b_2|} \right). \quad (24)$$

It only remains to find the value of the joint variables. Since all joints are revolute, their joint variables are angles. First, we need to construct four auxiliary lines and two planes with the already obtained points:

$$\ell_1^* = p_0 \wedge p_1 \wedge e_\infty, \quad (25)$$

$$\ell_2^* = p_1 \wedge p_2 \wedge e_\infty, \quad (26)$$

$$\ell_3^* = p_2 \wedge p_3 \wedge e_\infty, \quad (27)$$

$$\ell_4^* = p_3 \wedge p \wedge e_\infty, \quad (28)$$

$$\pi_1 = \mathbf{x}_1, \quad (29)$$

$$\pi_2^* = p_2 \wedge p_2 \wedge p_3 \wedge e_\infty. \quad (30)$$

Finally, using the geometric entities defined in Eqs. (15), (20) and (25–30), the joint variables are obtained:

$$\theta_1 = \angle(\pi_1^*, \pi^*), \quad (31)$$

$$\theta_2 = \angle(\ell_1^*, \ell_2^*), \quad (32)$$

$$\theta_3 = \angle(\ell_2^*, \ell_3^*), \quad (33)$$

$$\theta_4 = \angle(\ell_z^*, \pi_2^*), \quad (34)$$

$$\theta_5 = \angle(\ell_z^*, \ell_3^*), \quad (35)$$

$$\theta_6 = \angle(\ell_x^*, \ell_4^*), \quad (36)$$

where $\angle(\cdot, \cdot)$ denotes the main angle defined by two geometric entities. More precisely, for two geometric objects with outer representation K_1^* and K_2^* , it is defined by the following formula:

$$\angle(K_1^*, K_2^*) = \cos^{-1} \left(\frac{K_1^* \cdot K_2^*}{\|K_1^*\| \|K_2^*\|} \right). \quad (37)$$

3 Motion Planning

We start this section by giving some basic definitions that will allow us to formulate the motion planning problem for serial industrial robots [12, 16]. Here, a serial industrial robot of n DoF is denoted by \mathcal{R}_n .

Definition 6. The space of all positions and orientations that the end-effector of a serial industrial robot can reach is known as the *workspace* of the robot and is denoted by \mathcal{W} . Clearly, $\mathcal{W} \subset \mathcal{X}$.

Definition 7. An *obstacle* \mathcal{O} is a rigid-body object in \mathcal{W} .

Since obstacles are closed and bounded sets in \mathbb{R}^3 , they are compact sets with respect to the standard topology of \mathbb{R}^3 . Hence, every obstacle \mathcal{O} can be covered by a finite collection of open sets $\{S_j\}_{j=1}^m$, i.e., $\mathcal{O} \subset \bigcup_{j=1}^m S_j$. Obviously, since we are in \mathbb{R}^3 , these open sets may be assumed to be open balls. But then, we have that:

$$\mathcal{O} \subset \bigcup_{j=1}^m S_j \subset \bigcup_{j=1}^m \bar{S}_j, \quad (38)$$

so every obstacle \mathcal{O} can be covered by a finite set of closed balls.

As elements of the workspace of the robot, every obstacle can be represented in the configuration space \mathcal{C} .

Definition 8. Let \mathcal{R}_n be a serial robot and $\mathcal{O} \subset \mathcal{W}$, an obstacle. Then, as stated before, there exists a finite set of closed balls $\{\bar{S}_j\}_{j=1}^m$ such that $\mathcal{O} \subset \bigcup_{j=1}^m \bar{S}_j$. The representation of \mathcal{O} in the configuration space \mathcal{C} is computed as:

$$C(\mathcal{O}) = \{q \in \mathcal{C} : \mathcal{R}_n(q) \cap \bar{S}_j \neq \emptyset \text{ for some } j = 1, \dots, m\}, \quad (39)$$

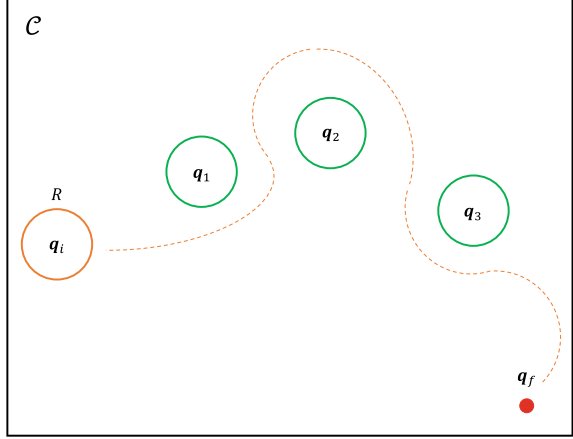
where $\mathcal{R}_n(q)$ is the subset of \mathcal{W} occupied by \mathcal{R}_n in that specific configuration q . Such representation is known as the *C-obstacle* or *C-obstacle* of \mathcal{O} .

Definition 9. Let \mathcal{R}_n be a serial robot and $\mathcal{O}_1, \dots, \mathcal{O}_r \subset \mathcal{W}$, r different obstacles. The *free-of-obstacles configuration space* C_{free} is:

$$C_{\text{free}} = \mathcal{C} \setminus \bigcup_{k=1}^r C(\mathcal{O}_k). \quad (40)$$

Proposition 1. The *free-of-obstacles configuration space* C_{free} is an open set.

Fig. 5 Graphical example of a planar motion planning problem: the mobile platform R should move from an initial configuration q_i to a goal configuration q_f avoiding the special configurations q_1 to q_3 in its way. A sphere is defined around each special configuration and, then, the solution trajectory is computed so its distance to all the spheres is always higher than a given value



Proof. We have seen that, for every $k = 1, \dots, r$, O_k is a closed set. In addition, since every serial industrial robot is a sequence of rigid-bodies, $\mathcal{R}_n(\mathbf{q})$ is also a closed set for every $\mathbf{q} \in C$ and, thus, $O_k \cap \mathcal{R}_n(\mathbf{q})$ is a closed set and so $C_{\text{obs}} = \bigcup_{k=1}^r C(O_k)$. Therefore, C_{free} is an open set.

Definition 10. The *motion planning problem* consists of finding a mapping, i.e., a path, $c : [0, 1] \rightarrow C$ such that no configuration along the path intersects an obstacle. If the map c has codomain C_{free} , the path is said to be a *free-path*.

Now, we can explore how conformal geometric algebra can be applied to this problem. In particular, it can be applied in two different ways: (1) by allowing an efficient computation of the free-of-obstacles configuration space C_{free} and (2) by improving the computation of the solution trajectories for any motion planning problem.

For the first application, the number and position of the closed balls covering each obstacle are not relevant. The only key point is the fact that they are covered by a finite set of closed balls (which is guaranteed by the compactness of the obstacles). Analogously, $\mathcal{R}_n(\mathbf{q})$ is also a compact set and, hence, can be covered by a finite set of closed balls. Therefore, we can rewrite Eq. (39) as:

$$C(O) = \left\{ \mathbf{q} \in C : \overline{C}_\ell \cap \overline{S}_j \neq \emptyset \text{ for } \begin{array}{l} \text{some } j = 1, \dots, m \\ \text{some } \ell = 1, \dots, p \end{array} \right\}, \quad (41)$$

where $\mathcal{R}_n(\mathbf{q}) \subset \bigcup_{\ell=1}^p \overline{C}_\ell$.

Now, the expression $\overline{C}_\ell \cap \overline{S}_j$ is equivalent to $(c_\ell \wedge s_j)^*$, where c_ℓ and s_j are the inner representation of the spheres defined by the boundary of \overline{C}_ℓ and \overline{S}_j respectively. Such intersection is empty if, and only if, $(c_\ell \wedge s_j)^2 > 0$. We add the extra condition $c_\ell \cdot s_j < 0$ since we want to avoid to have one sphere contained in the other (if $c_\ell \cdot s_j > 0$, the intersection between both spheres is still empty but one is contained

in the other and, hence, it is clear that such configuration does not belong to the free-of-obstacles configuration space). Therefore, Eq. (39) can be rewritten as:

$$C(O) = \left\{ \mathbf{q} \in C : \begin{array}{l} (c_\ell \wedge s_j)^2 > 0 \\ c_\ell \cdot s_j < 0 \end{array} \text{ for } \begin{array}{l} \text{some } j = 1, \dots, m \\ \text{some } \ell = 1, \dots, p \end{array} \right\}. \quad (42)$$

This is a more efficient way of computing $C(O)$ for every possible obstacle and, as a consequence, of computing C_{free} .

With respect to the second application, we should restrict the robots we use to serial industrial robots of two or three degrees of freedom so its configuration space has dimension two or three. This is done for practical reasons as we shall see later. This may seem too restrictive, but there is a large number of serial industrial robots with few degrees of freedom. A good example are mobile manipulators, where a platform with two translational degrees of freedom and one rotational degree of freedom moves freely in a planar environment (see Fig. 5). The main methods used for computing the solution trajectories in motion planning problems can be grouped into three categories:

- Potential field methods, where a differentiable real-valued function $U : C \rightarrow \mathbb{R}$, called the potential function, is defined. Such function has an attractive component that pulls the trajectory towards the goal configuration and a repulsive component that pushes the trajectory away from the start configuration and from the obstacles.
- Sampling-based multi-query methods, where a roadmap is constructed over C_{free} . The nodes represent free-of-obstacles configurations, while the edges represent feasible local paths between those configurations. Once the roadmap is constructed, a search algorithm finds out the best solution trajectory by selecting and joining the local paths through an optimization process.
- Sampling-based single-query methods, where a tree-structure data is constructed by searching new configurations (nodes) in C_{free} and connecting them through local paths (edges). Its main difference with respect to the multi-query methods is that, while the multi-query methods work in two times (construction of the roadmap and searching of a solution trajectory), in the single-query methods both steps are taken together. Each new configuration added to the set of nodes is connected by a local path and evaluated in order to check its feasibility.

Now, we will introduce a very useful concept. Given a sphere (or a circle) S centered at \mathbf{c} with radius $r > 0$ and a point \mathbf{p} exterior to S , we can always construct a right triangle with the segment defined by \mathbf{p} and \mathbf{c} as its hypotenuse and the radius as one of its legs. Then, by the Pythagorean theorem, the other leg has a length equal to $d^2(\mathbf{c}, \mathbf{p}) - r^2$. This is known as the *tangential distance of S from \mathbf{p}* and is denoted by $d_T(\mathbf{p}, S)$.

If s denotes the inner representation of S and \mathbf{p} and \mathbf{c} are the null vector representations of \mathbf{p} and \mathbf{c} , then:

$$\begin{aligned}
s \cdot p &= (c - \frac{1}{2}r^2 e_\infty) \cdot p \\
&= c \cdot p - \frac{1}{2}r^2 e_\infty \cdot p \\
&= -\frac{1}{2}d^2(c, p) + \frac{1}{2}r^2 \\
&= -\frac{1}{2}(d^2(c, p) - r^2) \\
&= -\frac{1}{2}d_T^2(p, S),
\end{aligned} \tag{43}$$

and, thus, $d_T(p, S) = \sqrt{-2s \cdot p}$.

For any of the three methods listed above, we can use the tangential distance d_T defined in Eq. (43) to simplify the computations and improve the overall performance. Indeed, let us first consider special configurations like, for instance, singularities—configurations where the serial industrial robot loses the ability to move in at least one direction of the operational space \mathcal{X} —and colliding configurations—configurations lying in the boundary between C_{free} and C_{obs} . These configurations should be taken into account when computing the solution trajectory of any motion planning problem. Then, for every special configuration q_0 , we define a sphere S centered at q_0 and with a small random radius $r > 0$. Now, we proceed as follows:

- For a potential field method, we define the potential function so it has a repulsive component that pushes the trajectory away from these special configurations. To do so, the most efficient way is to define, for each configuration q_0 of this type, a quadratic repulsive component as follows:

$$U_{q_0}(q) = \begin{cases} \frac{\kappa}{2} \left(\frac{1}{d_T(q, S)} - \frac{1}{d_0} \right)^2 & \text{if } d_T(q, S) \leq d_0 \\ 0 & \text{if } d_T(q, S) > d_0 \end{cases} \tag{44}$$

where s is the inner representation of S , d_0 is set as a threshold for the distance d_T and $\kappa \in \mathbb{R}$. Notice that $d_T^2(q, S) = -2q \cdot s$, where q is the null vector representation of q .

- For a sampling-based method with multiple queries, it is sufficient with removing from the roadmap those nodes associated with special configurations. During the construction of the roadmap, each configuration $q \in C$ is evaluated to determine whether q is free-of-obstacles or not. Similarly, the idea is to evaluate each $q \in C$ in order to determine whether q is close to a special configuration or not. To speed up the process, both evaluations can be carried out together:

- 1) Select a value $d_0 > 0$ that will work as a threshold.
- 2) Given a discretization of the configuration space C , each q of such discretization is evaluated to check whether:

- Is free-of-obstacles.
 - Is far from any special configuration \mathbf{q}_0 . This can be done simply by evaluating whether $d_T(\mathbf{q}, S) > d_0$ or $d_T(\mathbf{q}, S) \leq d_0$, where, again, s is the inner representation of the sphere S and $d_T^2(\mathbf{q}, S) = -2\mathbf{q} \cdot s$.
- 3) If \mathbf{q} is free-of-obstacles and far from any special configuration, then it can be added to the set of nodes of the roadmap.
- For a sampling-based method with a single query, the approach is completely analogous to the one used for methods with multiple-queries due to the similarities between both categories.

Remark 1. As stated before, we are restricting the robots we use to robots with two or three degrees of freedom. The reason stems from the fact that every method explained here works in the configuration space C of the robot. Hence, for an arbitrary serial industrial robot of n degrees of freedom, C has dimension n and any special configuration will be an n -dimensional vector $\mathbf{q} \in C$. Since the proposed solution is based on the computation of the tangential distance between the current robot configuration and several spheres (each one centered at one of those special configurations), we will need to compute the inner product between two grade-1 elements of $\mathcal{G}_{n+1,1}$. Theoretically speaking, there is no problem in that but, in practice, most of the current libraries, toolboxes and computer-based tools work with the conformal geometric algebra of vector spaces of small dimension, so the methods introduced here would not be able to be implemented.

4 Conclusions

In this chapter, we have dealt with three problems of serial industrial robots, namely the forward kinematics, the inverse kinematics and the motion planning problem. We have used the key elements of conformal geometric algebra to formally introduce them, as well as to develop specific tools to solve them. In particular, we have shown that given an arbitrary configuration $\mathbf{q} \in C$ of the robot, there exists a rotor $M \in \mathcal{G}_{4,1}$ such that $P' = MP\tilde{M}$ is either the position or orientation of the end-effector of the robot and where P is the position/orientation of the base frame. This means that rotor M relates the base frame with the end-effector's position and orientation for that specific configuration \mathbf{q} .

In addition, we have developed a geometric method based on the definition and manipulation of several geometric entities to solve the inverse kinematics of serial industrial robots without a spherical wrist and that, together with the method introduced in [13] provides a general method to deal with the inverse kinematics of arbitrary serial robots. Notice that the two methods have some similarities, so the question on whether a more general geometric method to solve the inverse kinematics of these robots exists or not arises naturally. Or whether is there a numerical method based on conformal geometric algebra that solves this problem? We have seen that,

in [1], a numerical method is developed using conformal geometric algebra. However, since its target is to solve just the inverse position problem of arbitrary long serial robots with only revolute joints, the algorithm can only be applied to these type of robots.

Finally, we have formulated the motion planning problem as well as shown how we can compute the free-of-obstacles configuration space C_{free} by means of evaluating the intersection between spheres when represented in conformal geometric algebra. In addition, we have also seen that we can compute a solution trajectory between two configurations $\mathbf{q}_i, \mathbf{q}_f \in C_{\text{free}}$ by evaluating, in each step of the process, the distance between the sphere or spheres representing the serial robot and the spheres covering each obstacle, singularity and/or forbidden configuration. There is, however, much to do in this field. For instance, two open questions are:

- We have seen several methods to compute solution trajectories. Can any of them be improved or its performance enhanced if entirely formulated using conformal geometric algebra?
- Could a new method be defined using this mathematical framework?

References

1. Aristidou, A., Lasenby, J.: Inverse kinematics solutions using conformal geometric algebra. In: Dorst, L., Lasenby, J. (eds.) *Guide to Geometric Algebra in Practice*, pp. 47–62. Springer, London (2011)
2. Baker, D.R., Wampler, C.W.: Some facts concerning the inverse kinematics of redundant manipulators. In: *IEEE International Conference on Robotics and Automation (ICRA)*, Raleigh, NC, USA, pp. 604–609, March 1987
3. Baker, D.R., Wampler, C.W.: On the inverse kinematics of redundant manipulators. *Int. J. Robot. Res.* **7**(2), 3–21 (1988)
4. Bayro-Corrochano, E., Zamora-Esquivel, J.: Differential and inverse kinematics of robot devices using conformal geometric algebra. *Robotica* **25**(1), 43–61 (2007)
5. Buss, S.R.: Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and Damped Least Squares methods. Technical report, University of California, San Diego (2009)
6. Campos-Macías, L., Carbajal-Espinosa, O., Loukianov, A., Bayro-Corrochano, E.: Inverse kinematics for a 6-DOF walking humanoid robot leg. *Adv. Appl. Clifford Algebras* **27**(1), 581–597 (2017)
7. Doran, C., Lasenby, A.: *Geometric Algebra for Physicists*. Cambridge University Press, Cambridge (2003)
8. Drexler, D.A.: Solution of the closed-loop inverse kinematics algorithm using the Crank-Nicolson method. In: *IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, Herlany, Slovakia, 21–23 January 2016, pp. 351–356 (2016)
9. Hildenbrand, D., Zamora, J., Bayro-Corrochano, E.: Inverse kinematics computation in computer graphics and robotics using conformal geometric algebra. *Adv. Appl. Clifford Algebras* **18**(3), 699–713 (2008)
10. Huang, L.S., Jiang, R.K.: A new method of inverse kinematics solution for industrial 7DoF robot. In: *32nd Chinese Control Conference (CCC)*, Xi'an, China, 26–28 July 2013, pp. 6063–6065 (2013)
11. Lasenby, A., Lasenby, J., Wareham, R.: A covariant approach to geometry using geometric algebra. Technical report, Department of Engineering, University of Cambridge (2004)

12. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
13. Latorre-Cugat, C., Xambó-Descamps, S., Zaplana, I.: A Geometric Algebra Invitation to Space-Time Physics, Robotics and Molecular Geometry. SBMAC/SpringerBriefs. Springer, Cham (2018)
14. Paul, R.P.: Robot Manipulators: Mathematics, Programming and Control. The MIT Press, Cambridge (1981)
15. Pieper, D.L.: The kinematics of manipulation under computer control. Ph.D. thesis, Stanford Artificial Intelligence Laboratory - Stanford University (1968)
16. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics: Modelling, Planning and Control. Springer, London (2008)
17. Spong, M.W., Hutchinson, S., Vidyasagar, M.: Robot Modeling and Control. Wiley, Hoboken (2006)
18. Tørdal, S.S., Hovland, G., Tyapin, I.: Efficient implementation of inverse kinematics on a 6-DOF industrial robot using conformal geometric algebra. *Adv. Appl. Clifford Algebras* **27**(3), 2067–2082 (2017)
19. Zamora, J., Bayro-Corrochano, E.: Inverse kinematics, fixation and grasping using conformal geometric algebra. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004, vol. 4, pp. 3841–3846 (2004)