19. Kinematics of Robot Manipulators in the Motor Algebra*

Eduardo Bayro-Corrochano and Detlef Kähler

Institute of Computer Science and Applied Mathematics,
Christian-Albrechts-University of Kiel

19.1 Introduction

In the literature we find a variety of mathematical approaches for solving problems in robotics which we will review now briefly. Denavit and Hartenberg [60] introduced the mostly used kinematic notation for lower pair mechanisms based on matrix algebra, Walker [243] used the epsilon algebra for the treatment of the manipulator kinematics, Gu and Luh [99] utilized dual–matrices for computing the Jacobians useful for kinematics and robot dynamics and Pennock and Yang [188] derived closed–form solutions for the inverse kinematics problem for various types of robot manipulators employing dual–matrices. McCarthy [171] used the dual form of the Jacobian for the analysis of multi–links similarly. Funda and Paul [87] gave a detailed computational analysis of the use of screw transformations in robotics. These authors explained that since the dual quaternion can represent the rotation and translation transformations simultaneously it is more effective than the unit quaternion formalism for dealing with the kinematics of robot chains. Kim and Kumar [132] computed a closed–form solution of the inverse kinematics of a 6 degree of freedom robot manipulator in terms of line transformations using dual quaternions. Aspragathos and Dimitros [9] confirmed once again

* This work has been supported by DFG Grant So-320-2-1.

that the use of dual quaternion and Lie algebra in robotics were overseen so far and that their use helps to reduce the number of representation parameters.

We can see in all these mathematical approaches that the authors take into account basically two key aspects: the obvious use of dual numbers and the representation of the screw transformations in terms of matrices or dual quaternions. In this regard in this chapter we are concerned with the extension of the representation capabilities of the dual numbers, particularly using the motor algebra beside the point and line representation we are able to model the motion of planes. This widens up the possibilities for the modelling of the motion of the basic geometric objects referred to frames attached to the robot manipulator which according to the circumstances simplify the complexity of the problem preserving the underlying geometry. After giving the modelling of prismatic and revolute transformations of a robot manipulator using points, lines and planes we solve the direct and inverse kinematics of robot manipulators. Using the motion of points, lines and planes in terms of motors we present constraints for a simple grasping task. The chapter shows clearly the advantages of the use of representations in motor algebra for solving problems related to robot manipulators.

The organization of the chapter is as follows: section two describes the prismatic and revolute transformations of robot manipulators in the motor algebra framework. The third section deals with the computation of the direct kinematics of robot manipulators. The fourth section is dedicated to the solution of the inverse kinematics of one standard robot manipulator. Finally, section five presents the conclusions.

19.2 Motor Algebra for the Kinematics of Robot Manipulators

The study of the rigid motion of objects in 3D space plays an important role in robotics. In order to linearize the rigid motion of the Euclidean space homogeneous coordinates are normally utilized. That is why in the geometric algebra framework we choose the special or degenerated geometric algebra to extend the algebraic system from 3D Euclidean space to the 4D space. In this system we can nicely model the motion of points, lines and planes with computational advantages and geometric insight, see chapter 18 for more details. Let us start with a description of the basic elements of robot manipulators in terms of the special or degenerated geometric algebra $\mathcal{G}^{+}_{3,0,1}$ or motor algebra. The most basic parts of a robot manipulator are revolute joints, prismatic joints, connecting links and the end–effectors. In the next subsections we will treat the kinematics of the prismatic and revolute manipulator parts using the 4D geometric algebra $\mathcal{G}^{+}_{3,0,1}$ and will illustrate an end–effector grasping task.

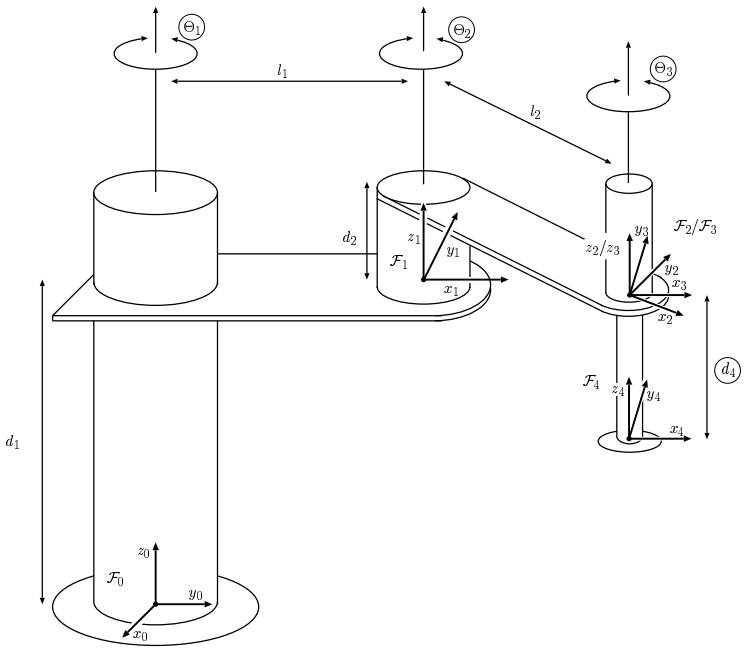

Fig. 19.1a. SCARA type manipulator according to the DH parameters in table 19.1. Variable parameters are encircled

19.2.1 The Denavit–Hartenberg Parameterization

The computation of the direct or inverse kinematics requires both the exact description of the robot manipulators structure and its configuration. The mostly used description approach is known as Denavit–Hartenberg procedure [60]. This is based on the uniform description of the position of the reference coordinate system of a joint relative to the next one in consideration. Figure 19.2a shows how coordinate frames are attached to a joint of a robot manipulator. Table 19.1 presents the specifications of two robot manipulators: the SCARA and the Stanford manipulator as shown in figures 19.1a and 19.1b, respectively.

In table 19.1 a variable parameter is indicated by the letter v and a constant one by c. This tells us whether the joint is for rotation (revolute) or for translation (prismatic). The transformation of the reference coordinate system between two joints will be called joint–transition . Figure 19.2b shows the involved screws in a joint–transition according to the Denavit–Hartenberg parameters . The frame or reference coordinate system related to the i-th joint is attached at the end of this link and it is called $\mathcal{F}_i$. The position and orientation of the end–effector in relation to the reference coordinate system of the robot basis can be computed by linking all joint–transitions. In this

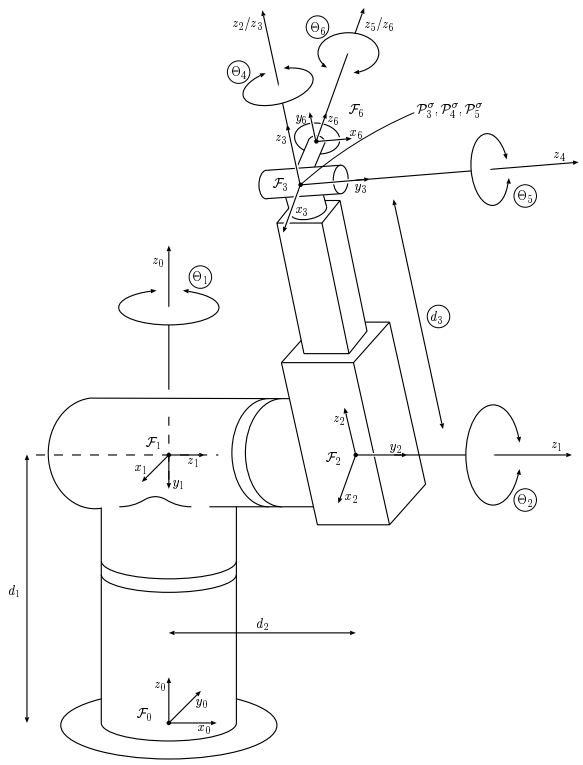

Fig. 19.1b. Stanford type manipulator according to the DH parameters in table 19.1. Variable parameters are encircled

way we get straightforwardly the direct kinematics.

Conversely for the inverse kinematics given the position and orientation of the end–effector we have to find values of the variable parameters of the joint–transitions which satisfy this requirement. In the next sections we will go more into details about the computation of direct and inverse kinematics of robot manipulators.

19.2.2 Representations of Prismatic and Revolute Transformations

The transformation of any point , line or plane between coordinate systems $\mathcal{F}_{i-1}$ and $\mathcal{F}_i$ is a revolute one when the degree of freedom is only a variable angle θ_i and a prismatic one when the degree of freedom is only a variable length d_i. The transformation motor ${}^{i-1}\boldsymbol{M}_i$ between $\mathcal{F}_i$ and $\mathcal{F}_{i-1}$ consists of a sequence of two screw transformations , one fixed, i.e. $\boldsymbol{M}^x_{\hat{\alpha}_i}$, and another variable , i.e. $\boldsymbol{M}^z_{\hat{\theta}_i}$, see figure 19.2b. Note that we use dual angles $\hat{\theta}_i = \theta_i + Id_i$ and $\hat{\alpha}_i = \alpha_i + I\hat{l}_i$, see chapter 18. In the revolute case the latter has as a variable parameter the angle θ_i and in the prismatic case the displacement d_i. The transformation reads

Table 19.1. Kinematic configuration of two robot manipulators

Robot type	link	revolute θ_i	v/c	prismatic d_i	v/c	twist angle α_i	link length l_i
SCARA	1	θ_1	v	d_1	c	0	l_1
	2	θ_2	v	d_2	c	0	l_2
	3	θ_3	v	0		0	0
	4	0		d_4	v	0	0
Stanford	1	θ_1	v	d_1	c	-90 deg	0
	2	θ_2	v	d_2	c	90 deg	0
	3	0		d_3	v	0	0
	4	θ_4	v	0		-90 deg	0
	5	θ_5	v	0		90 deg	0
	6	θ_6	v	d_6	c	0	0

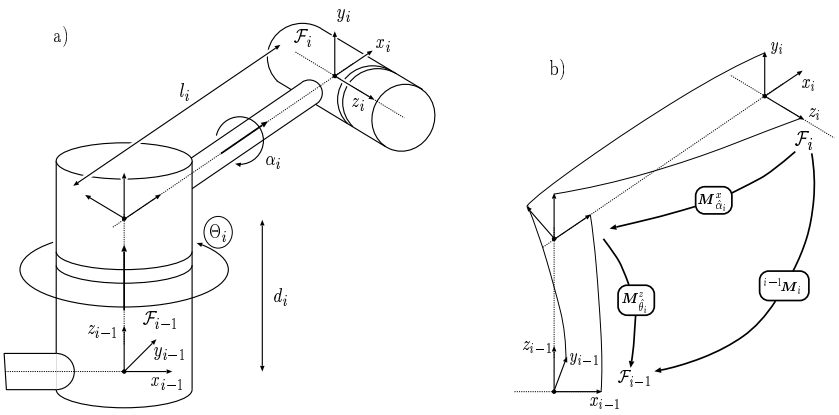

Fig. 19.2. a) The i-th joint of a robot manipulator and the attached coordinate frames according to the Denavit–Hartenberg procedure. Here the encircled θ_i is the variable parameter, b) the transformation from frame $\mathcal{F}_i$ to $\mathcal{F}_{i-1}$ is represented by ${}^{i-1}\boldsymbol{M}_i$. The motor ${}^{i-1}\boldsymbol{M}_i$ consists of two screw transformations $\boldsymbol{M}^x_{\hat{\alpha}_i}$ and $\boldsymbol{M}^z_{\hat{\theta}_i}$

$$
\begin{aligned}
{}^{i-1}\boldsymbol{M}_i &= \boldsymbol{M}^z_{\hat{\theta}_i}\boldsymbol{M}^x_{\hat{\alpha}_i} = \boldsymbol{T}^z_{d_i}\boldsymbol{R}^z_{\theta_i}\boldsymbol{T}^x_{l_i}\boldsymbol{R}^x_{\alpha_i} \\
&= (1+\frac{I}{2}\begin{pmatrix}0\\0\\d_i\end{pmatrix})\boldsymbol{R}^z_{\theta_i}(1+\frac{I}{2}\begin{pmatrix}l_i\\0\\0\end{pmatrix})\boldsymbol{R}^x_{\alpha_i}\ . \qquad (19.1)
\end{aligned}
$$

For the sake of clearness the dual bivectors of translators are given as a column vector simply to make the variable parameters explicit.

Since ${}^{i-1}\boldsymbol{M}_i\ {}^{i-1}\widetilde{\boldsymbol{M}}_i = 1$, we obtain

$$
{}^{i}\boldsymbol{M}_{i-1} = \widetilde{\boldsymbol{M}}^x_{\hat{\alpha}_i}\widetilde{\boldsymbol{M}}^z_{\hat{\theta}_i} = \widetilde{\boldsymbol{T}}^x_{l_i}\widetilde{\boldsymbol{R}}^x_{\alpha_i}\widetilde{\boldsymbol{T}}^z_{d_i}\widetilde{\boldsymbol{R}}^z_{\theta_i}\ . \qquad (19.2)
$$

Be aware for the rest of the chapter that ${}^{j}\boldsymbol{M}_i$ denotes a motor transformation from $\mathcal{F}_i$ to $\mathcal{F}_j$.

We will now give general expressions for the transformation of points, lines and planes with one of the parameters θ_i and d_i, respectively, as a variable and with two fixed parameters α_i and l_i. In the joint depicted in figure 19.2b a revolute transformation will take place only when θ_i varies and a prismatic transformation only when d_i varies. Now taking a point $\boldsymbol{X}$ represented in the frame $\mathcal{F}_{i-1}$, we can describe its transformation from $\mathcal{F}_{i-1}$ to $\mathcal{F}_i$ in the motor algebra according to chapter 18 with either θ_i or d_i as variable parameter. We will call this transformation a *forward transformation* .

The multivector representation of point $\boldsymbol{X}$ related to the frame $\mathcal{F}_i$ will be expressed as ${}^{i}\boldsymbol{X}$ with

$$\begin{aligned} {}^{i}\boldsymbol{X} &= {}^{i}\boldsymbol{M}_{i-1}\ {}^{i-1}\boldsymbol{X}\ {}^{i}\widetilde{\overline{\boldsymbol{M}}}_{i-1} = \widetilde{\boldsymbol{M}}^{x}_{\hat{\alpha}_i}\widetilde{\boldsymbol{M}}^{z}_{\hat{\theta}_i}\ {}^{i-1}\boldsymbol{X}\ \overline{\boldsymbol{M}}^{z}_{\hat{\theta}_i}\overline{\boldsymbol{M}}^{x}_{\hat{\alpha}_i} \\ &= \widetilde{\boldsymbol{T}}^{x}_{l_i}\widetilde{\boldsymbol{R}}^{x}_{\alpha_i}\widetilde{\boldsymbol{T}}^{z}_{d_i}\widetilde{\boldsymbol{R}}^{z}_{\theta_i}\ {}^{i-1}\boldsymbol{X}\ \boldsymbol{R}^{z}_{\theta_i}\widetilde{\boldsymbol{T}}^{z}_{d_i}\boldsymbol{R}^{x}_{\alpha_i}\widetilde{\boldsymbol{T}}^{x}_{l_i} \\ &= 1 + I\,{}^{i}\boldsymbol{x}\ , \end{aligned} \tag{19.3}$$

where ${}^{i}\boldsymbol{x}$ is a bivector representing the 3D position of $\boldsymbol{X}$ referred to $\mathcal{F}_i$. Thinking in a transformation in the reverse sense we call it a *backward transformation* which transforms a point $\boldsymbol{X}$ represented in the frame $\mathcal{F}_i$ to the frame $\mathcal{F}_{i-1}$ as follows

$$\begin{aligned} {}^{i-1}\boldsymbol{X} &= {}^{i-1}\boldsymbol{M}_i\ {}^{i}\boldsymbol{X}\ {}^{i-1}\widetilde{\overline{\boldsymbol{M}}}_i = \boldsymbol{M}^{z}_{\hat{\theta}_i}\boldsymbol{M}^{x}_{\hat{\alpha}_i}\ {}^{i}\boldsymbol{X}\ \widetilde{\overline{\boldsymbol{M}}}^{x}_{\hat{\alpha}_i}\ \widetilde{\overline{\boldsymbol{M}}}^{z}_{\hat{\theta}_i} \\ &= 1 + I\,{}^{i-1}\boldsymbol{x}\ . \end{aligned} \tag{19.4}$$

Note that the motor applied from the right side is not purely conjugated as in the line case. This will be also the case for a plane, see chapter 18 for details of the point and plane transformations.

Consider a line $\boldsymbol{L}$ represented in the frame $\mathcal{F}_{i-1}$ by ${}^{i-1}\boldsymbol{L} = {}^{i-1}\boldsymbol{n} + I^{i-1}\boldsymbol{m}$, where $\boldsymbol{n}$ and $\boldsymbol{m}$ are bivectors indicating the orientation and moment of the line, respectively. We can write its forward transformation related to the frame $\mathcal{F}_i$ according to chapter 18 as follows

$$\begin{aligned} {}^{i}\boldsymbol{L} &= {}^{i}\boldsymbol{M}_{i-1}\ {}^{i-1}\boldsymbol{L}\ {}^{i}\widetilde{\boldsymbol{M}}_{i-1} = \widetilde{\boldsymbol{M}}^{x}_{\hat{\alpha}_i}\widetilde{\boldsymbol{M}}^{z}_{\hat{\theta}_i}\ {}^{i-1}\boldsymbol{L}\ \boldsymbol{M}^{z}_{\hat{\theta}_i}\boldsymbol{M}^{x}_{\hat{\alpha}_i} \\ &= {}^{i}\boldsymbol{n} + I\,{}^{i}\boldsymbol{m}\ . \end{aligned} \tag{19.5}$$

Its backward transformation reads

$$\begin{aligned} {}^{i-1}\boldsymbol{L} &= {}^{i-1}\boldsymbol{M}_i\ {}^{i}\boldsymbol{L}\ {}^{i-1}\widetilde{\boldsymbol{M}}_i = \boldsymbol{M}^{z}_{\hat{\theta}_i}\boldsymbol{M}^{x}_{\hat{\alpha}_i}\ {}^{i}\boldsymbol{L}\ \widetilde{\boldsymbol{M}}^{x}_{\hat{\alpha}_i}\widetilde{\boldsymbol{M}}^{z}_{\hat{\theta}_i} \\ &= {}^{i-1}\boldsymbol{n} + I\,{}^{i-1}\boldsymbol{m}\ . \end{aligned} \tag{19.6}$$

Finally, the forward transformation of a plane $\boldsymbol{H}$ represented in $\mathcal{F}_{i-1}$ reads

$$\begin{aligned} {}^{i}\boldsymbol{H} &= {}^{i}\boldsymbol{M}_{i-1}\ {}^{i-1}\boldsymbol{H}\ {}^{i}\widetilde{\overline{\boldsymbol{M}}}_{i-1} = \widetilde{\boldsymbol{M}}^{x}_{\hat{\alpha}_i}\widetilde{\boldsymbol{M}}^{z}_{\hat{\theta}_i}\ {}^{i-1}\boldsymbol{H}\ \overline{\boldsymbol{M}}^{z}_{\hat{\theta}_i}\overline{\boldsymbol{M}}^{x}_{\hat{\alpha}_i} \\ &= {}^{i}\boldsymbol{n} + I\,{}^{i}d_H\ . \end{aligned} \tag{19.7}$$

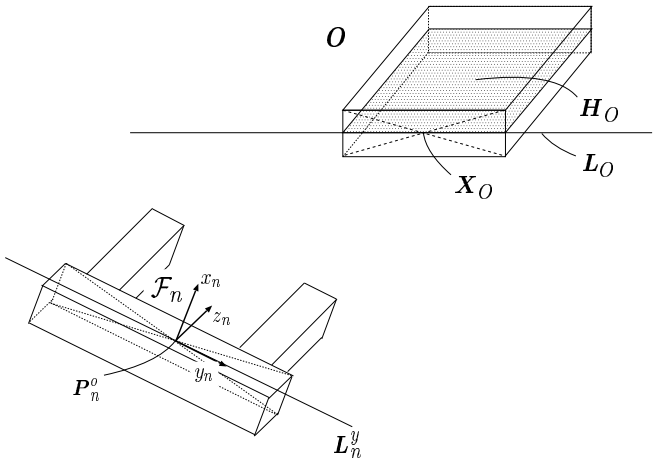

Fig. 19.3. Two finger grasper approaching to an object

and similarly as above, its backward transformation equation is

$$\begin{aligned} {}^{i-1}\boldsymbol{H} &= {}^{i-1}\boldsymbol{M}_i \; {}^{i}\boldsymbol{H} \; {}^{i-1}\widetilde{\overline{\boldsymbol{M}}}_i = \boldsymbol{M}^z_{\hat{\theta}_i} \boldsymbol{M}^x_{\hat{\alpha}_i} \; {}^{i}\boldsymbol{H} \; \widetilde{\overline{\boldsymbol{M}}}^x_{\hat{\alpha}_i} \; \widetilde{\overline{\boldsymbol{M}}}^z_{\hat{\theta}_i} \\ &= {}^{i-1}\boldsymbol{n} + I \, {}^{i-1}d_H \; . \end{aligned} \tag{19.8}$$

19.2.3 Grasping by Using Constraint Equations

In this subsection we will illustrate grasping as a manipulation related task. grasping operation. This task involves the positioning of a two finger grasper in front of a static object. Figure 19.3 shows the grasper and the considered object $\boldsymbol{O}$. The manipulator moves the grasper near to the object and together they should fulfill some conditions to grasp the object firmly. In order to determine the overall transformation ${}^0\boldsymbol{M}_n$, which moves the grasper to an appropriate grasping position, we claim that ${}^0\boldsymbol{M}_n$ has to fulfill three constraints. For the formulation of these constraints we can take advantages of the point, line and plane representations of the motor algebra. In the following we assume that the representations of geometric entities attached to the object $\boldsymbol{O}$ in frame $\mathcal{F}_0$ are known.

Attitude condition: The grasping movement of the two fingers should be in the reference plane $\boldsymbol{H}_O$ of $\boldsymbol{O}$. That is, the yz-plane of the end–effector frame $\mathcal{F}_n$ should be equal to the reference plane $\boldsymbol{H}_O$. The attitude condition can be simply formulated in terms of a plane equation as follows

$${}^0\boldsymbol{M}_n \; {}^n\boldsymbol{H}^{yz}_n \; {}^0\widetilde{\overline{\boldsymbol{M}}}_n - {}^0\boldsymbol{H}_O \approx 0 \; , \tag{19.9}$$

where ${}^n\boldsymbol{H}^{yz}_n = (1,0,0)^T + I\,0 = (1,0,0)^T$, see figure 19.3.

Alignment condition: The grasper and object should be aligned parallel after the application of the motor ${}^0\boldsymbol{M}_n$. That is, the direction of the y-axis

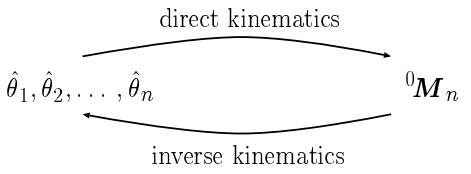

Fig. 19.4. Direct and inverse kinematics

and the line $\boldsymbol{L}_O$ should be the same. This condition can be simply expressed in terms of a line equation

$$\langle {}^0\boldsymbol{M}_n \, {}^n\boldsymbol{L}_n^y \, {}^0\widetilde{\boldsymbol{M}}_n \rangle_d - \langle {}^0\boldsymbol{L}_O \rangle_d \approx 0 \; , \tag{19.10}$$

where ${}^n\boldsymbol{L}_n^y = (0,1,0)^T + I(0,0,0)^T = (0,1,0)^T$ and $\langle \boldsymbol{L} \rangle_d$ denotes the components of direction of line $\boldsymbol{L}$.

Touching condition: The motion ${}^0\boldsymbol{M}_n$ should also guarantee that the grasper is in the right grasping position. That is, the origin $\boldsymbol{P}_n^o$ of the end–effector frame $\mathcal{F}_n$ should touch the reference point $\boldsymbol{X}_O$ of $\boldsymbol{O}$. A formulation of this constraint in our framework is

$${}^0\boldsymbol{M}_n \, {}^n\boldsymbol{P}_n^o \, {}^0\widetilde{\boldsymbol{M}}_n - {}^0\boldsymbol{X}_O \approx 0 \; . \tag{19.11}$$

By these three conditions we get constraints for the components of ${}^0\boldsymbol{M}_n$, and we can determine ${}^0\boldsymbol{M}_n$ numerically. The next step is to determine the variable joint parameters of the robot manipulator which leads to the position and orientation of the end–effector frame $\mathcal{F}_n$ described by ${}^0\boldsymbol{M}_n$. This problem is called the inverse kinematics problem of robot manipulators and will be treated in section 19.4.

19.3 Direct Kinematics of Robot Manipulators

The direct kinematics involves the computation of the position and orientation of the end–effector or frame $\mathcal{F}_n$ given the parameters of the joint–transitions, see figure 19.4. In this section we will show how the direct kinematics can be computed when we use as geometric object a point, line or plane. The notation for points, lines and planes we will use in the next sections is illustrated in figure 19.5. The direct kinematics for the general case of a manipulator with n joints can be written as follows

$${}^0\boldsymbol{M}_n = {}^0\boldsymbol{M}_1 \, {}^1\boldsymbol{M}_2 \, {}^2\boldsymbol{M}_3 \cdots {}^{n-1}\boldsymbol{M}_n = \prod_{i=1}^{n} {}^{i-1}\boldsymbol{M}_i \; . \tag{19.12}$$

Now we can formulate straightforwardly the direct kinematics in terms of

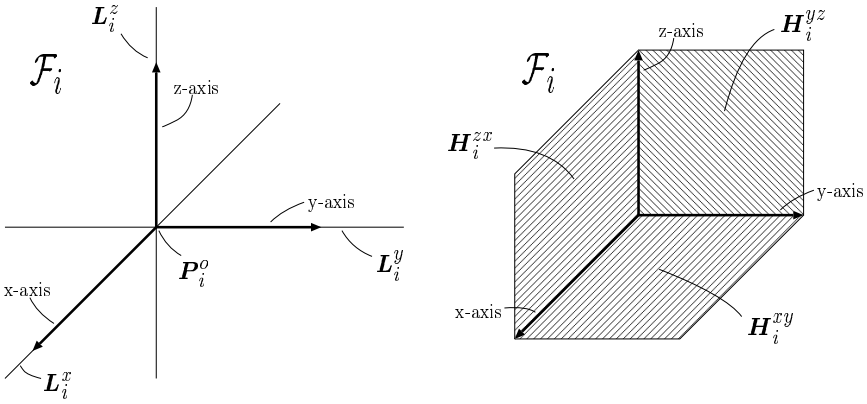

Fig. 19.5. Notations for frame specific entities as the origin, the coordinate axis and coordinate planes

point, line or plane representations as follows

$$
\begin{aligned}
{}^0\boldsymbol{X} &= {}^0\boldsymbol{M}_n\ {}^n\boldsymbol{X}\ {}^0\widetilde{\overline{\boldsymbol{M}}}_n = \prod_{i=1}^{n} {}^{i-1}\boldsymbol{M}_i\ {}^n\boldsymbol{X}\ \prod_{i=1}^{n} {}^{n-i}\widetilde{\overline{\boldsymbol{M}}}_{n+1-i}\ , \\
{}^0\boldsymbol{L} &= \prod_{i=1}^{n} {}^{i-1}\boldsymbol{M}_i\ {}^n\boldsymbol{L}\ \prod_{i=1}^{n} {}^{n-i}\widetilde{\boldsymbol{M}}_{n+1-i}\ , \\
{}^0\boldsymbol{H} &= \prod_{i=1}^{n} {}^{i-1}\boldsymbol{M}_i\ {}^n\boldsymbol{H}\ \prod_{i=1}^{n} {}^{n-i}\widetilde{\overline{\boldsymbol{M}}}_{n+1-i}\ .
\end{aligned}
\tag{19.13}
$$

Let us now write the motor ${}^0\boldsymbol{M}_4$ for the direct kinematics for points, lines and planes like equation (19.13) for the SCARA manipulator specified by the Denavit–Hartenberg parameters of table 19.1. Firstly, using equation (19.12) with n=4, we can write down straightforwardly the required motor ${}^0\boldsymbol{M}_4$ as follows

$$
\begin{aligned}
{}^0\boldsymbol{M}_4 &= {}^0\boldsymbol{M}_1\,{}^1\boldsymbol{M}_2\,{}^2\boldsymbol{M}_3\,{}^3\boldsymbol{M}_4 = (\boldsymbol{M}^z_{\hat{\theta}_1}\boldsymbol{M}^x_{\hat{\alpha}_1})\cdots(\boldsymbol{M}^z_{\hat{\theta}_4}\boldsymbol{M}^x_{\hat{\alpha}_4}) \\
&= (\boldsymbol{T}^z_{d_1}\boldsymbol{R}^z_{\theta_1}\boldsymbol{T}^x_{l_1}\boldsymbol{R}^x_{\alpha_1})\cdots(\boldsymbol{T}^z_{d_4}\boldsymbol{R}^z_{\theta_4}\boldsymbol{T}^x_{l_4}\boldsymbol{R}^x_{\alpha_4}) \\
&= (1+\frac{I}{2}\begin{pmatrix}0\\0\\d_1\end{pmatrix})\boldsymbol{R}^z_{\theta_1}(1+\frac{I}{2}\begin{pmatrix}l_1\\0\\0\end{pmatrix})(1+\frac{I}{2}\begin{pmatrix}0\\0\\d_2\end{pmatrix}) \\
&\quad \boldsymbol{R}^z_{\theta_2}(1+\frac{I}{2}\begin{pmatrix}l_2\\0\\0\end{pmatrix})\boldsymbol{R}^z_{\theta_3}(1+\frac{I}{2}\begin{pmatrix}0\\0\\d_4\end{pmatrix})\ .
\end{aligned}
\tag{19.14}
$$

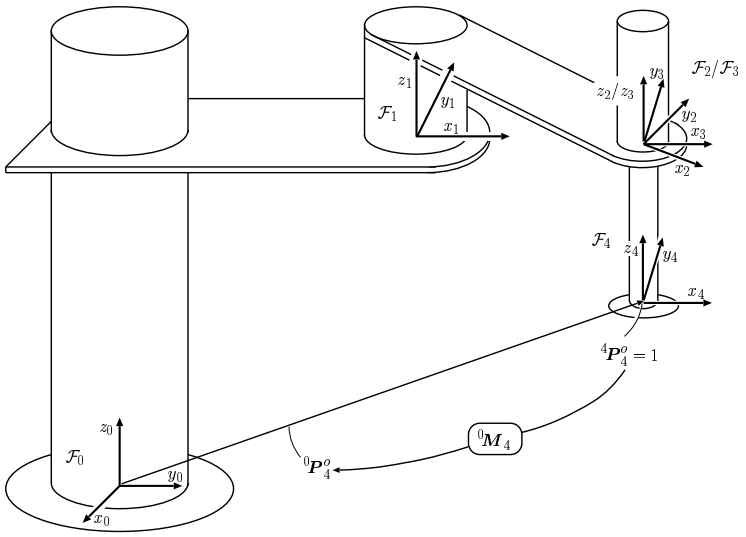

Fig. 19.6. The representation ${}^0\boldsymbol{P}_4^o$ of $\boldsymbol{P}_4^o$ in frame $\mathcal{F}_0$ is computed using ${}^0\boldsymbol{M}_4$

Note that translators with zero translation and rotors with zero angle become 1.

Applying the motor ${}^0\boldsymbol{M}_4$ from the left and ${}^0\widetilde{\overline{\boldsymbol{M}}}_4$ from the right for point and plane equations and the motor ${}^0\boldsymbol{M}_4$ from the left and ${}^0\widetilde{\boldsymbol{M}}_4$ from the right for line equations as indicated by equations (19.13), we get the direct kinematics equations of points, lines and planes for the SCARA robot manipulator.

19.3.1 Maple Program for Motor Algebra Computations

Since the nature of our approach requires symbolic computation we chose Maple to implement a program suitable for computations in the motor algebra framework $\mathcal{G}^+_{3,0,1}$. We have developed a comfortable program for computations in the frame of different geometric algebras. When dealing with the motor algebra we have simply to specify its vector basis. The program has a variety of useful algebraic operators to carry out computations involving reversion, Clifford conjugations, inner and wedge operations, rotations, translations, motors, extraction of the i–blade of a multivector etc.

As a first illustration using our Maple program, we computed the direct kinematic equation of the origin $\boldsymbol{P}_4^o$ of $\mathcal{F}_4$ for the SCARA manipulator specified by the Denavit–Hartenberg parameters of table 19.1. The figure 19.6

shows the frames and the point $\boldsymbol{P}_4^o$ refered to $\mathcal{F}_0$. The final result is

$$
\begin{aligned}
{}^0\boldsymbol{P}_4^o &= {}^0\boldsymbol{M}_4 \; {}^4\boldsymbol{P}_4^o \; {}^0\widetilde{\boldsymbol{M}}_4 = {}^0\boldsymbol{M}_4 \left(1 + I \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}\right) {}^0\widetilde{\boldsymbol{M}}_4 \\
&= 1 + I \begin{pmatrix} l_2 \cos(\theta_1 + \theta_2) + l_1 \cos(\theta_1) \\ l_2 \sin(\theta_1 + \theta_2) + l_1 \sin(\theta_1) \\ d_1 + d_2 + d_4 \end{pmatrix} . \qquad (19.15)
\end{aligned}
$$

19.4 Inverse Kinematics of Robot Manipulators

Since the inverse kinematics is more complex than the direct kinematics our aim should be to find a systematic way to solve it exploiting the point, line and plane motor algebra representations. Unfortunately the procedure is not amenable for a general formulation as in the case of the direct kinematics equation (19.12). That is why we better choose a real robot manipulator and compute its inverse kinematics in order to show all the characteristics of the computational assumptions.

The Stanford robot manipulator is well known among researchers concerned with the design of strategies for the symbolic computation of the inverse kinematics. According to table 19.1 the variable parameters to be computed are θ_1, θ_2, θ_4, θ_5, θ_6 and d_3. By means of this example we will show that in the motor algebra approach we have the freedom to switch between the point, line or plane representation according to the geometrical circumstances. This is one of the most important advantages of our motor algebra approach.

According to the mechanical characteristics of the Stanford manipulator we can divide it into two basic parts: one dedicated for the positioning involving the joints 1,2 and 3 and one dedicated for the orientation of the end–effector like a wrist comprising the joints 4 to 6. Since the philosophy of our approach relies on the application of point, line or plane representation where it is needed, we should firstly recognize whether a point or a line or a plane representation is the suitable representation for the joint–transitions. As a result on the one hand a better geometric insight is guaranteed and on the other hand the solution method is easier to be developed. The first three joints of the Stanford manipulator are used to position the origin of the coordinate frame $\mathcal{F}_3$. Therefore we apply a point representation to describe this part of the problem. The last three joints are used to achieve to desired orientation of the end–effector frame. For the formulation of this subproblem we use a line and a plane representation because with these entities we can model orientations.

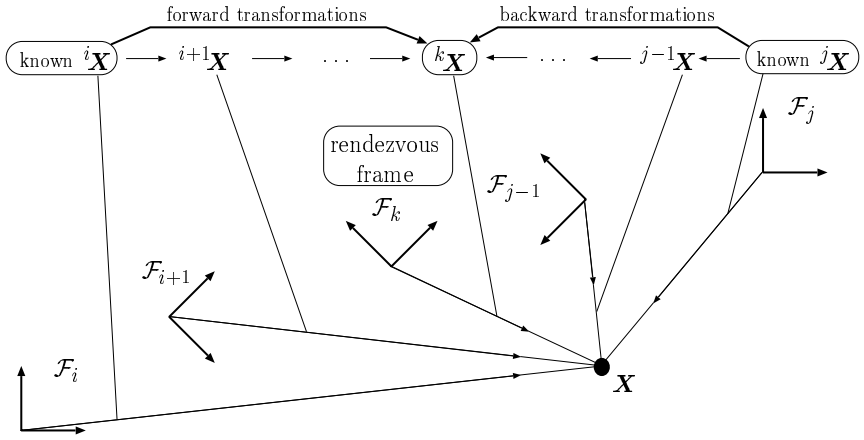

Fig. 19.7. Rendezvous method: If ${}^{i}\boldsymbol{X}$ and ${}^{j}\boldsymbol{X}$ are known, we can compute ${}^{k}\boldsymbol{X}$ for each $i \leq k \leq j$ in two different ways: by successive forward transformations of ${}^{i}\boldsymbol{X}$ and by successive backward transformation of ${}^{j}\boldsymbol{X}$

19.4.1 The Rendezvous Method

The next important step is to represent the motor transformations from the beginning of a chain of joint–transitions to the end and vice versa as it is depicted in figure 19.7. As a result we gain a set of equations for each meeting point. In each of these points the forward equation is equal with the backward equation. Using these equalities we have a guideline to compute the unknowns. We will call this procedure the *rendezvous method* . This simple idea has proved to be very useful as a strategy for the solution of the inverse kinematics.

19.4.2 Computing θ_1, θ_2 and d_3 Using a Point Representation

In the case of the Stanford manipulator the orientation and position of frame $\mathcal{F}_6$ uniquely determines the position of frame $\mathcal{F}_3$. This will be explained in the following.

The position of frame $\mathcal{F}_3$ with respect to $\mathcal{F}_0$ is described by the multivector representation ${}^{0}\boldsymbol{P}_3^o$ of $\boldsymbol{P}_3^o$ in $\mathcal{F}_0$. By successive forward transformation applied on ${}^{3}\boldsymbol{P}_3^o = 1$ we get the representation ${}^{6}\boldsymbol{P}_3^o$ of $\boldsymbol{P}_3^o$ in $\mathcal{F}_6$ by

$$ {}^{6}\boldsymbol{P}_3^o = {}^{6}\boldsymbol{M}_3 \; {}^{3}\boldsymbol{P}_3^o \; {}^{6}\widetilde{\overline{\boldsymbol{M}}}_3 = 1 - I \begin{pmatrix} 0 \\ 0 \\ d_6 \end{pmatrix} . \tag{19.16} $$

Now we can compute ${}^{0}\boldsymbol{P}_3^o$ by

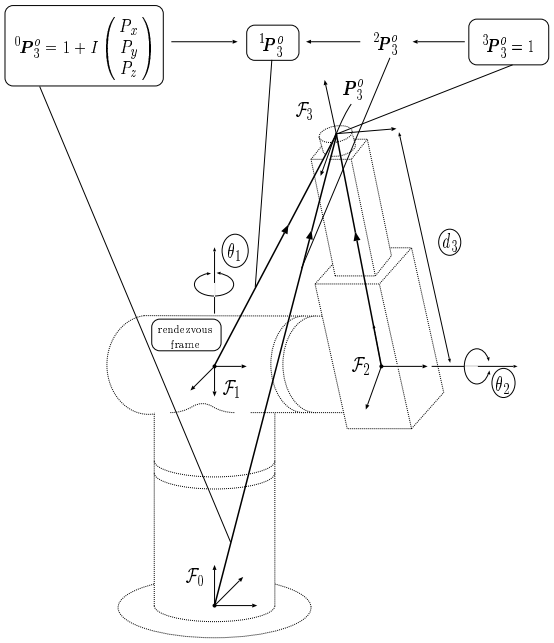

Fig. 19.8. The rendezvous method applied to $\boldsymbol{P}_3^o$ in order to determine the equations shown in table 19.2. The equations of rendezvous frame $\mathcal{F}_1$ are choosen to compute the variable parameters θ_1, θ_2 and d_3

$$
\begin{aligned}
{}^0\boldsymbol{P}_3^o &= {}^0\boldsymbol{M}_6 \; {}^6\boldsymbol{P}_3^o \; {}^0\widetilde{\boldsymbol{M}}_6 = {}^0\boldsymbol{M}_6 \; (1 - I \begin{pmatrix} 0 \\ 0 \\ d_6 \end{pmatrix}) \; {}^0\widetilde{\boldsymbol{M}}_6 \\
&= 1 + I \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} \; , \qquad (19.17)
\end{aligned}
$$

note that ${}^0\boldsymbol{M}_6$ is given. The vector $(P_x, P_y, P_z)^T$ describes the position of the origin $\boldsymbol{P}_3^o$ of frame $\mathcal{F}_3$ in frame $\mathcal{F}_0$ for a given overall transformation ${}^0\boldsymbol{M}_6$. Now we can apply the rendezvous method since we know the representation of $\boldsymbol{P}_3^o$ in the two different frames $\mathcal{F}_0$ and $\mathcal{F}_3$, see figure 19.8.

Applying successive forward transformations we obtain

$$
\begin{aligned}
{}^1\boldsymbol{P}_3^o &= {}^1\boldsymbol{M}_0 \; {}^0\boldsymbol{P}_3^o \; {}^1\widetilde{\boldsymbol{M}}_0 \; , \\
{}^2\boldsymbol{P}_3^o &= {}^2\boldsymbol{M}_1 \; {}^1\boldsymbol{P}_3^o \; {}^2\widetilde{\boldsymbol{M}}_1 \; , \\
{}^3\boldsymbol{P}_3^o &= {}^3\boldsymbol{M}_2 \; {}^2\boldsymbol{P}_3^o \; {}^3\widetilde{\boldsymbol{M}}_2 \; . \qquad (19.18)
\end{aligned}
$$

These computations were carried out with our Maple program getting the left hand sides of the four groups of equations of the table 19.2.

On the other hand, applying successive backward transformations to the origin of $\mathcal{F}_3$ given by

$$ {}^3\boldsymbol{P}_3^o = 1 + I \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = 1 \,, \tag{19.19} $$

we get

$$ {}^2\boldsymbol{P}_3^o = {}^2\boldsymbol{M}_3 \; {}^3\boldsymbol{P}_3^o \; {}^2\widetilde{\boldsymbol{M}}_3 = 1 + I \begin{pmatrix} 0 \\ 0 \\ d_3 \end{pmatrix} \,, $$

$$ {}^1\boldsymbol{P}_3^o = {}^1\boldsymbol{M}_2 \; {}^2\boldsymbol{P}_3^o \; {}^1\widetilde{\boldsymbol{M}}_2 = 1 + I \begin{pmatrix} d_3 \sin(\theta_2) \\ -d_3 \cos(\theta_2) \\ d_2 \end{pmatrix} \,, $$

$$ {}^0\boldsymbol{P}_3^o = {}^0\boldsymbol{M}_1 \; {}^1\boldsymbol{P}_3^o \; {}^0\widetilde{\boldsymbol{M}}_1 = 1 + I \begin{pmatrix} d_3 \sin(\theta_2)\cos(\theta_1) - d_2 \sin(\theta_1) \\ d_3 \sin(\theta_2)\sin(\theta_1) + d_2 \cos(\theta_1) \\ d_3 \cos(\theta_2) + d_1 \end{pmatrix} \tag{19.20} $$

These equations correspond to the right hand sides of the four groups of equations of table 19.2. For simplicity we use the abbreviations s_i for $\sin(\theta_i)$ and c_i for $\cos(\theta_i)$. Using the third equation of the rendezvous frame $\mathcal{F}_1$, we

Table 19.2. Rendezvous equations obtained for $\boldsymbol{P}_3^o$ regarding frames $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2$ and $\mathcal{F}_3$

Frame	Eq.	forward		backward
	1	P_x	=	$d_3 s_2 c_1 - d_2 s_1$
$\mathcal{F}_0$	2	P_y	=	$d_3 s_2 c_1 + d_2 c_1$
	3	P_z	=	$d_3 c_2 + d_1$
	1	$P_y s_1 + P_x c_1$	=	$d_3 s_2$
$\mathcal{F}_1$	2	$d_1 - P_z$	=	$-d_3 c_2$
	3	$P_y c_1 - P_x s_1$	=	d_2
	1	$-P_z s_2 + d_1 s_2 + P_x c_1 c_2 + P_y s_1 c_2$	=	0
$\mathcal{F}_2$	2	$d_2 - P_y c_1 + P_x s_1$	=	0
	3	$P_z c_2 - d_1 c_2 + P_x c_1 s_2 + P_y s_1 s_2$	=	d_3
	1	$-P_z s_2 + d_1 s_2 + P_x c_1 c_2 + P_y s_1 c_2$	=	0
$\mathcal{F}_3$	2	$d_2 - P_y c_1 + P_x s_1$	=	0
	3	$P_z c_2 - d_1 c_2 + P_x c_1 s_2 + P_y s_1 s_2 - d_3$	=	0

compute

$$\theta_1 = \arctan_2(x_{1/2}, y_{1/2}) \,, \tag{19.21}$$

where

$$x_{1/2} = \frac{d_2 - P_y y_{1/2}}{-P_x}, \qquad y_{1/2} = \frac{P_y d_2 \pm P_x \sqrt{P_x^2 + P_y^2 - d_2^2}}{P_x^2 + P_y^2} \tag{19.22}$$

and

$$\arctan_2(x,y) = \begin{cases} \arctan(\frac{x}{y}) & : \quad y > 0 \\ \frac{\pi}{2} & : \quad y = 0 \text{ and } x > 0 \\ \text{undefined} & : \quad y = 0 \text{ and } x = 0 \\ -\frac{\pi}{2} & : \quad y = 0 \text{ and } x < 0 \\ \arctan(\frac{x}{y}) + \pi & : \quad y < 0 \end{cases} . \tag{19.23}$$

This gives two values for θ_1. Now let us look for d_3 and θ_2. For that we consider the first and second equation of the rendezvous frame $\mathcal{F}_1$. With $a_{1/2} = P_y x_{1/2} + P_x y_{1/2}$ and $b = P_z - d_1$ we get two values for d_3. Since for the Stanford manipulator d_3 must be positive, we choose

$$d_{3_{1/2}} = \sqrt{a_{1/2}^2 + b^2} \,. \tag{19.24}$$

Using this value in equations 1 and 2, we compute straightforwardly

$$\theta_2 = \arctan_2(\frac{a_{1/2}}{d_{3_{1/2}}}, \frac{b}{d_{3_{1/2}}}) \,. \tag{19.25}$$

19.4.3 Computing θ_4 and θ_5 Using a Line Representation

These variables will be computed using the joint–transition from $\mathcal{F}_3$ to $\mathcal{F}_6$. According to the geometric characteristics of the manipulator it appears appealing that we should use the line representation to set up an appropriate equation system. The representation ${}^0\boldsymbol{L}_6^z$ of the line $\boldsymbol{L}_6^z$ in frame $\mathcal{F}_0$ can be computed using ${}^0\boldsymbol{M}_6$

$${}^0\boldsymbol{L}_6^z = {}^0\boldsymbol{M}_6 \; {}^6\boldsymbol{L}_6^z \; {}^0\widetilde{\boldsymbol{M}}_6 = {}^0\boldsymbol{M}_6 \left(\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + I \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right) {}^0\widetilde{\boldsymbol{M}}_6 \,. \tag{19.26}$$

Since the z-axis of $\mathcal{F}_6$ frame crosses the origin of $\mathcal{F}_3$, we can see that the z-axis line related to this frame has zero moment. Thus we can claim that L_6^z in $\mathcal{F}_3$ frame is

$$ {}^3\boldsymbol{L}_6^z = {}^3\boldsymbol{M}_0 \; {}^0\boldsymbol{L}_6^z \; {}^3\widetilde{\boldsymbol{M}}_0 = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} + I \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} . \tag{19.27} $$

Note that ${}^3\boldsymbol{M}_0$ is known since we have already computed θ_1, θ_2 and d_3.

Now applying successively forward transformations as follows

$$ \begin{aligned} {}^4\boldsymbol{L}_6^z &= {}^4\boldsymbol{M}_3 \; {}^3\boldsymbol{L}_6^z \; {}^4\widetilde{\boldsymbol{M}}_3 \; , \\ {}^5\boldsymbol{L}_6^z &= {}^5\boldsymbol{M}_4 \; {}^4\boldsymbol{L}_6^z \; {}^5\widetilde{\boldsymbol{M}}_4 \; , \\ {}^6\boldsymbol{L}_6^z &= {}^6\boldsymbol{M}_5 \; {}^5\boldsymbol{L}_6^z \; {}^6\widetilde{\boldsymbol{M}}_5 \; , \end{aligned} \tag{19.28} $$

we get the left hand sides of the four groups of equations of table 19.3. The z-axis line $\boldsymbol{L}_6^z$ of $\mathcal{F}_6$ represented in $\mathcal{F}_6$ has zero moment, thus it can be expressed as

$$ {}^6L_6^z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + I \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} . \tag{19.29} $$

Now applying successive backward transformations, we have

$$ \begin{aligned} {}^5\boldsymbol{L}_6^z &= {}^5\boldsymbol{M}_6 \; {}^6\boldsymbol{L}_6^z \; {}^5\widetilde{\boldsymbol{M}}_6 \; , \\ {}^4\boldsymbol{L}_6^z &= {}^4\boldsymbol{M}_5 \; {}^5\boldsymbol{L}_6^z \; {}^4\widetilde{\boldsymbol{M}}_5 \; , \\ {}^3\boldsymbol{L}_6^z &= {}^3\boldsymbol{M}_4 \; {}^4\boldsymbol{L}_6^z \; {}^3\widetilde{\boldsymbol{M}}_4 \; . \end{aligned} \tag{19.30} $$

Using our Maple program, we compute the right hand sides of the four groups of equations of table 19.3. We will consider the equations of rendezvous frame

Table 19.3. Rendezvous equations obtained for $\boldsymbol{L}_6^z$ regarding frames $\mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5$ and $\mathcal{F}_6$

Frame	Eq.	forward		backward
$\mathcal{F}_3$	1	A_x	=	$-c_4 s_5$
	2	A_y	=	$-s_4 s_5$
	3	A_z	=	$-c_5$
$\mathcal{F}_4$	1	$A_y s_4 + A_x c_4$	=	$-s_5$
	2	A_z	=	$-c_5$
	3	$A_y c_4 - A_x s_4$	=	0
$\mathcal{F}_5$	1	$-A_z s_5 + A_x c_4 c_5 + A_y s_4 c_5 c_6$	=	0
	2	$A_y c_4 - A_x s_4$	=	0
	3	$-A_z c_5 - A_x c_4 s_5 - A_y s_4 s_5$	=	1
$\mathcal{F}_6$	1	$A_x s_4 s_6 - A_y c_4 s_6 + A_y s_4 c_5 c_6 + A_x c_4 c_5 c_6 - A_z s_5 c_6$	=	0
	2	$-A_x s_4 c_6 + A_y c_4 c_6 + A_y s_4 c_5 c_6 + A_x c_4 c_5 s_6 - A_z s_5 s_6$	=	0
	3	$-A_z c_5 - A_x c_4 s_5 - A_y s_4 s_5$	=	1

$\mathcal{F}_4$. Using the third equation, we compute

$$\theta_4 = \arctan_2(x_{1/2}, y_{1/2}) \,, \tag{19.31}$$

where

$$x_{1/2} = -\frac{A_y y_{1/2}}{-Ax} = \pm\frac{A_y}{\sqrt{A_x^2 + A_y^2}} \,, \qquad y_{1/2} = \pm\frac{A_x}{\sqrt{A_x^2 + A_y^2}} \,. \tag{19.32}$$

This results in two values for θ_4 which substituted in the first and second equation helps us to find two solutions for θ_5

$$\theta_5 = \arctan_2(s_5, c_5) = \arctan_2\Big((-A_y s_4 - A_x c_4), -A_z\Big) \,. \tag{19.33}$$

19.4.4 Computing θ_6 Using a Plane Representation

Since θ_1, θ_2, d_3, θ_4 and θ_5 are now known, we can compute the motor ${}^5\boldsymbol{M}_0$. The yz–plane $\boldsymbol{H}_6^{yz}$ represented in $\mathcal{F}_6$ has the Hesse distance 0, thus

$${}^6\boldsymbol{H}_6^{yz} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + I0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \,. \tag{19.34}$$

Its transformation to $\mathcal{F}_0$ reads

$${}^0\boldsymbol{H}_6^{yz} = {}^0\boldsymbol{M}_6 \; {}^6\boldsymbol{H}_6^{yz} \; {}^0\widetilde{\overline{\boldsymbol{M}}}_6 = {}^0\boldsymbol{M}_6 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} {}^0\widetilde{\overline{\boldsymbol{M}}}_6 \,. \tag{19.35}$$

Now we compute ${}^5\boldsymbol{H}_6^{yz}$ by

$${}^5\boldsymbol{H}_6^{yz} = {}^5\boldsymbol{M}_0 \; {}^0\boldsymbol{H}_6^{yz} \; {}^5\widetilde{\overline{\boldsymbol{M}}}_0 = \begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix} + I \, {}^5d_{H_6^{yz}} \,. \tag{19.36}$$

The orientation bivector $(N_x, N_y, N_z)^T$ describes the orientation of the yz-plane of frame $\mathcal{F}_6$ in frame $\mathcal{F}_5$ given the values of the joint variables $\theta_1, \theta_2, \theta_4, \theta_5$ and d_3. Now applying forward transformation from $\mathcal{F}_5$ to $\mathcal{F}_6$, we obtain

$${}^6\boldsymbol{H}_6^{yz} = {}^6\boldsymbol{M}_5 \; {}^5\boldsymbol{H}_6^{yz} \; {}^6\widetilde{\overline{\boldsymbol{M}}}_5 \,. \tag{19.37}$$

Using our Maple program, we get the left hand sides of the two groups of equations of the table 19.4. Since the values for θ_1, θ_2, d_3, θ_4 and θ_5 are not

Table 19.4. Rendezvous equations obtained for $\boldsymbol{H}_6^{yz}$ regarding frames $\mathcal{F}_5$ and $\mathcal{F}_6$

Frame	Eq.	forward		backward
	1	N_x	=	c_6
$\mathcal{F}_5$	2	N_y	=	s_6
	3	N_z	=	0
	1	$N_y s_6 + N_x c_6$	=	1
$\mathcal{F}_6$	2	$N_x s_6 - N_y c_6$	=	0
	3	N_z	=	0

unique we, will get different values for the equations. Applying ${}^5\boldsymbol{M}_6$ to ${}^6\boldsymbol{H}_6^{yz}$ we get, the right hand sides of the two groups of equations of table 19.4 by

$$ {}^5\boldsymbol{H}_6^{yz} = {}^5\boldsymbol{M}_6 \; {}^6\boldsymbol{H}_6^{yz} \; {}^5\widetilde{\overline{\boldsymbol{M}}}_6 = {}^5\boldsymbol{M}_6 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \; {}^5\widetilde{\overline{\boldsymbol{M}}}_6 = \begin{pmatrix} \sin(\theta_6) \\ \cos(\theta_6) \\ 0 \end{pmatrix} . \quad (19.38) $$

We will consider the equations of the rendezvous frame $\mathcal{F}_5$. Using the first and second equation, we can compute θ_6 by

$$ \theta_6 = \arctan_2(s_6, c_6) = \arctan_2(N_x, N_y) \; . \quad (19.39) $$

Note that since we had two values for θ_4 and two values for θ_5, there is more than one solution for θ_6.

19.5 Conclusion

This chapter presented the application of the algebra of motors for the treatment of the direct and inverse kinematics of robot manipulators. When dealing with 3D rigid motion it is usual to use homogeneous coordinates in the 4D space to linearize this non–linear 3D transformation. With the same effect we model the prismatic and revolute motion of points, lines and planes using motors which are equivalent to screws. The fact that in our approach we can also use the representation of planes widens up the geometric language for the treatment of robotic problems.

The chapter has shown the flexibility of the motor algebra approach for the solution of the direct and inverse kinematics of robot manipulators. Using a standard robot manipulator, we show that according to the need we can resort for solving its inverse kinematics either to a point, a line or a plane representation. Thus, the main contribution of this chapter is to show that while preserving the geometric insight during the computation our approach gains more flexibility. The authors of this chapter believe that the increasing complexity of future multi–links mechanisms will profit from the versatility of the motor algebra framework.